100 words per minute, is the smallest teleprinter in the world in production, has a smoothly styled modern appearance, and is easily convertible to either 5-wire or sequential operation.

This machine is shown in Figure 4. It has been in production for some months and is the basic unit of an integrated range of telegraph machines that are now in various stages of development. A reperforating attachment for the model-Seventy-five teleprinter is already in production, whilst a new automatic tape transmitter attachment, a printing reperforator, and a tape teleprinter are on the way.

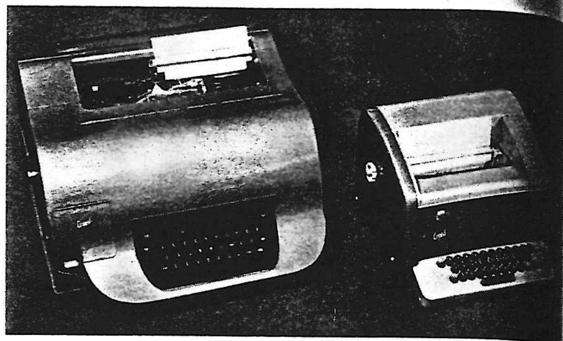


Figure 4—The model-Seventy-five teleprinter at the right is only ab the size and a third of the weight of former models. It is capable of op at 100 words per minute with greatly extended servicing intervals.

Mention of the model-Seventy-five teleprinter brings us to the present phase of Creed de ment. As this is, naturally, of greater in than the largely historical material in this ar a full, separate article has been devoted to

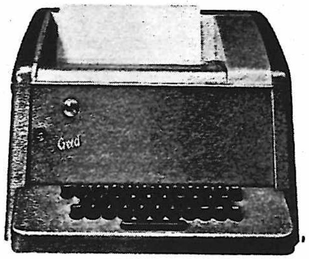
---

Creed Model-Seventy-Five Teleprinter

By BRUCE BROOKE-WAVELL

Creed & Company, Limited; Croydon, England

EARLY IN 1958, the first production units of a new teleprinter left the Creed factory. This model Seventy-five machine (Figure 1) is the smallest teleprinter in production in the world, and its weight of 35 pounds (16 kilograms) for the receiver-only version is well under half that of most other teleprinters.



Considerable interest in these features was shown by aircraft operating companies well before the production stage was reached. In March, 1956, a pre-production version of the machine was fitted as an airborne teleprinter in a British Overseas Airways Corporation Stratocruiser on regular service from London to New York and continuously recorded weather data and other essential flight information broadcast from radio stations in Galdenoch, Scotland, and Halifax, Nova Scotia. This trial was very successful, the operation of the machine being unaffected by either vibration or tilting.

All this might suggest, on the principle that one cannot get something for nothing, that small size and light weight have been obtained at the expense of robustness, range of facilities, standards of performance, or other features. This, however, is not so. It is the first and basic unit of a new integrated range of teleprinter equipment to supersede existing machines, both in conventional telegraph communication services and in the growing number of special applications in data processing and automation. The surprising reductions in size and weight are byproducts of new design methods.

The basic design of the model-Seventy-five teleprinter is quite different from that of present machines. Existing Creed teleprinters were developed by modifying and adapting the model 7, which was originally introduced in 1931, to provide the greater variety of facilities and higher standards of performance called for by progress in telegraphy since that time. Although these machines are adequate for most present needs, it is becoming increasingly difficult to add new facilities to them without also adding disproportionately to their size and weight. By making a completely fresh start, it has been possible to provide the full range of present-day facilities in a much-simpler and more-direct manner while, at the same time, making full allowance for foreseeable future requirements.

1. *General Features*

The teleprinter has been designed to operate reliably at 100 words per minute to cater for an increasingly widespread requirement, especially in data processing and computer output applications. At the more-commonly used communication speeds of 66 and 60 words per minute, to which the teleprinter can be adapted by a simple gear change, this reserve speed provides a large additional safety factor.

Single- and double-current operation are both catered for, the minimum receiver operating current being 40 and 20 milliamperes, respectively. The same transmitter contact assembly, without requiring any relay, is used for both modes of operation. The 5 code elements are first set up simultaneously on 5 changeover contacts and then transmitted sequentially by 5 make–break contacts. This use of separate transmitter contacts for each code element gives optimum transmission characteristics on single-current circuits.

The basic operating facility provided is simplex plus local record, the local record being obtained by a direct mechanical connection between the keyboard and receiver and not, as on previous machines, electrically through a leak resistor. The transmitter changeover contacts read off the transmitted combinations from the selecting pins on the receiver and the make–break contacts are

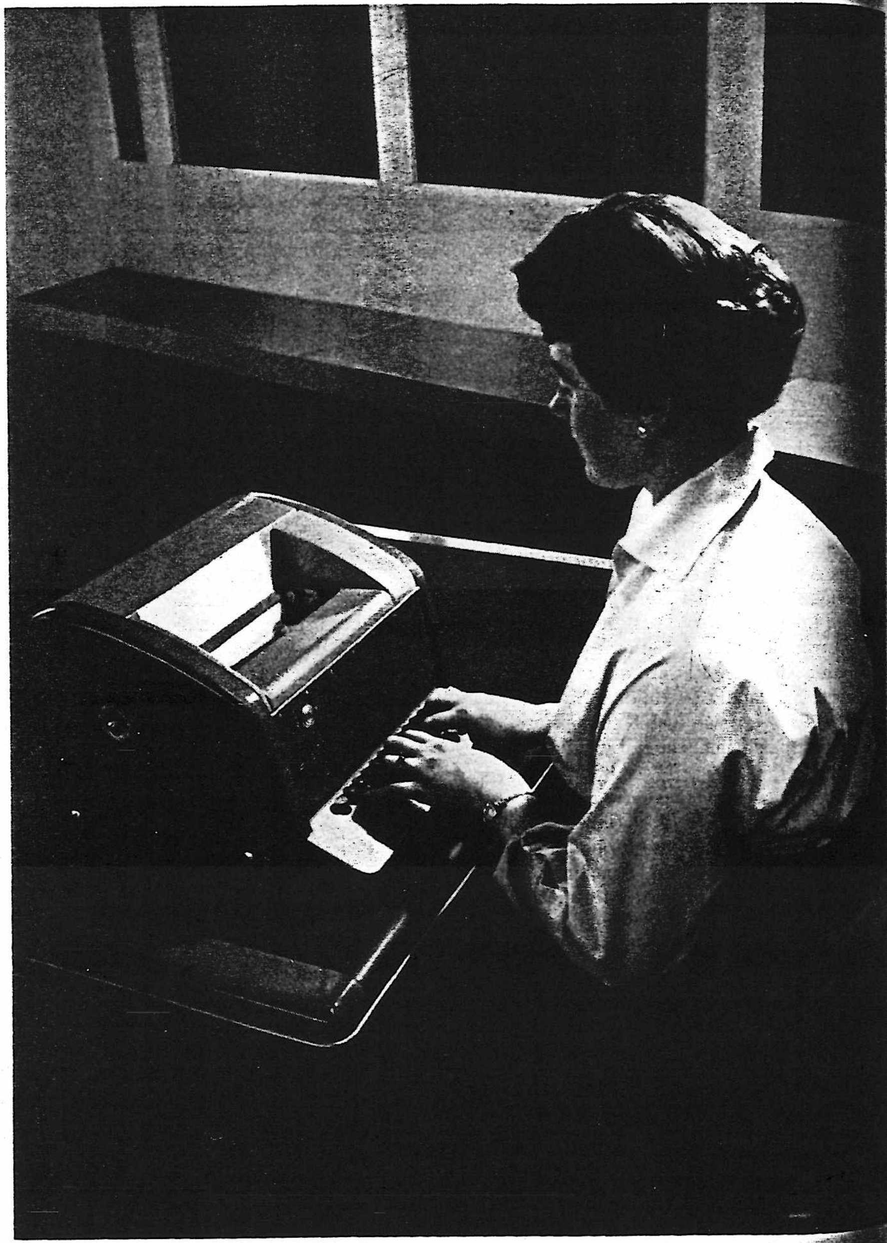


Figure 1—Smaller, lighter, and simpler than any other teleprinter in production in the world, the model Seve
is capable of sustained operation at 100 words per minute with high standards of attention-free performa

operated by cams on the receive cam sleeve. The need for a separate keyboard transmitter is thereby avoided. This feature is responsible for a substantial reduction in the number of parts in the teleprinter. A further advantage is that the local record is printed immediately after the depression of a key, without any perceptible time lag.

The keyboard can be of the 3- or 4-row type, and the several hundred keyboard layouts catered for on previous Creed teleprinters are all available on the new machine.

The teleprinter is constructed on the unit principle, which has been proved by experience to result in simpler maintenance and manufacture. All units are interchangeable with equivalent units on other machines provided they are adjusted.

Special care has been taken to reduce the maintenance required; the lubrication intervals, for example, being extended to approximately 1000 hours of operation, compared with 300 hours for previous machines. This has been achieved by improved machine movements and by the more extensive use of self-lubricating bearings and felt washers.

Attention has been given to making the cover of the machine both functionally efficient and pleasing in appearance. It is aluminium with smoothly styled lines and a distinctive two-tone grey-silver hammer finish. As part of the sound-reducing arrangements, the cover is made to enclose the machine completely and is lined with an acoustic material, easy access to the paper and ink ribbon being provided by hinged rear and window sections.

2. *Design Principles*

The teleprinter receiver is fitted with a single driving shaft, powered by a lightweight 4200-revolution-per-minute, fractional-horsepower motor. This shaft drives the selector and translator cam shafts through friction clutches. The use of a single driving shaft has the advantage of keeping distortion due to gears to a minimum, whilst the use of friction clutches ensures a constant pick-up time, which contributes to the receiver margin.

The selector cam shaft controls the selector mechanism, which converts the signal combination registered by the receiving electromagnet into a code setting on one or other of two sets of 5 pins on the translator unit. The method of selection employed secures an operating margin of ±40 per cent at a telegraph speed of 50 bauds and ±35 per cent at 75 bauds. An orientation adjustment is provided by a simple device that enables the selecting actions to be made earlier or later with respect to the start pulses of incoming signals by any percentage of the length of a signal element up to ±70 per cent. This provides a simple means of checking the margin of the receiver and enables the selector mechanism to be set in the centre of the tolerance range.

The translator cam shaft controls the translator mechanism. Its main function is to convert the selecting-pin code setting into the appropriate machine action, that is into printing a character or performing a function. It has been mentioned that there are actually two sets of 5 pins. This duplication enables a combination to be set up by the selector mechanism on one set of pins at the same time that the previous combination, which has already been set up on the other set of pins, is being read by the translator. The box holding the two sets of pins is made to move forwards and backwards by the translator unit so that each set of pins engages alternately with the selector and translator mechanism. This mode of operation permits the teleprinter to print each combination immediately after selection without storing it until the receipt of the following combination.

The translator unit effects the printing of the selected characters by moving a typewheel through a simple link-type aggregate-motion mechanism (Figure 2). The typewheel is much smaller and lighter than previous Creed typewheels (see Figure 3 for a comparison between the new typewheel and that on the model-*54* teleprinter). It is mounted vertically and has 4 layers of type on it, one under the other, each layer containing 16 types. It has four degrees of movement:—

A. *Lateral movement* by a rack-and-pawl mechanism actuated by a cam in the translator unit. The typewheel traverses the length of the platen, a character at a time, except when a functional combination is received. If this combination is the carriage-return signal, the typewheel returns sharply to the beginning of the line, where it is smoothly brought to rest by a small piston-type dashpot under the main base.

B. *Vertical movement* under dual control of the letters–figures-shift mechanism and element *2* of the received code combination. In the rest position, to which the typewheel returns after each character has been printed, the top of the typewheel is below the level of the printing point (see Figure 4), thereby ensuring immedia complete visibility. When the translator shaft rotates, the typewheel is raised amount depending on whether the selected acter is in the letters or figures case and w element *2* is a mark or space. One of 4 la types is thus raised level with the pr point. (The mechanical details are illustra Figure 2.)

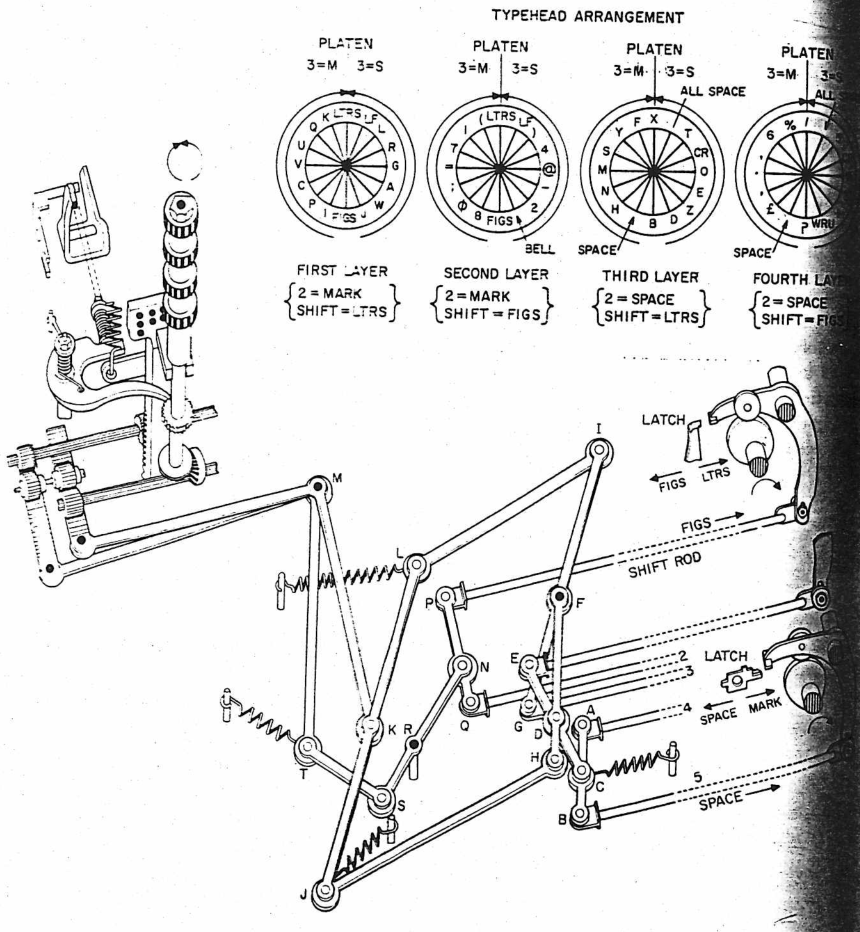


Figure 2—Aggregate-motion-mechanism diagram. Pivots *M*, *R*, and *F* are fixed. Rod movements for shift and *1*, *2*, *3*, and *4* are 0·5 inch (127 millimetres) and for element *5* is 0·25 inch (64 millimetres). Proportions of le *AC/CB* = 1/1, *ED/DC* = 1/1, *FD/DH* = 3/1, *IF/FG* = 4/3, *LK/KJ* = 1/1, and *PN/NQ* = 2/1.

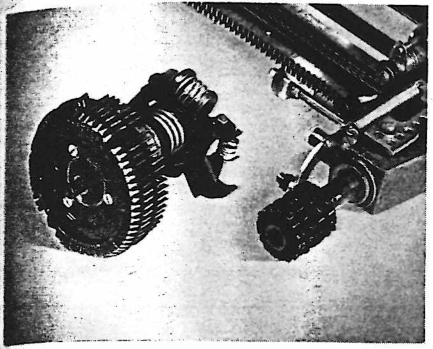


Figure 3A—Comparison of typehead units on the model Seventy-five (right) and model-*54* (left) teleprinters.

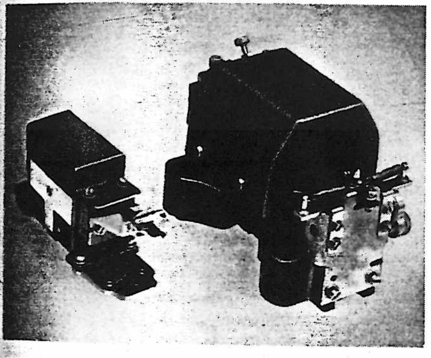


Figure 3C—Comparison of electromagnet units for the model Seventy-five (right) and model *54* (left).

C. *Rotary movement* under control of elements *1*, *3*, *4*, and *5*, of the received code combination. The positions of the selecting pins for these elements determine which of the corresponding control rods on the aggregate-motion mechanism should move to the right (see Figure 2). This determines both the direction and degree of rotation of the typewheel, the direction of rotation being controlled by element *3* and the angle of rotation, which never exceeds half a revolution, being controlled by the combined action of elements *1*, *4*, and *5*. The proportions of the levers in the aggregate-motion mechanism are so arranged that for each combination of marks and spaces for the elements *1*, *3*, *4*, and *5*, one of the 16 types on the selected layer is brought opposite the printing point.

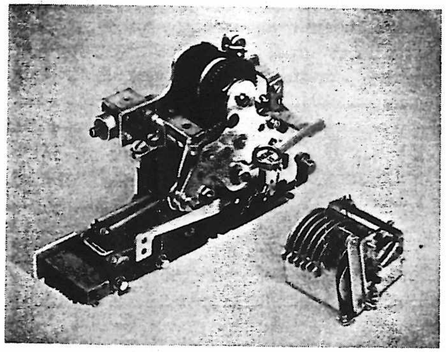
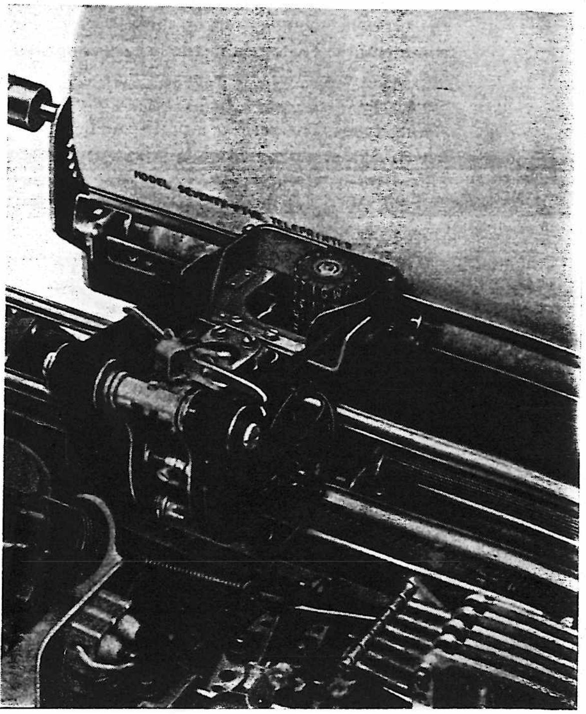


Figure 3B—Comparison of answer-back units on the model Seventy-five (right) and model *54* (left).

D. *Forward movement* by the printing mechanism that is also energised by a cam on the translator unit. After the selected character is positioned

Figure 4—Close-up of printing point showing stationary platen and moving typehead. The typehead returns instantly at all speeds, operation being unaffected by shock, tilting, or vibration. Every character is printed immediately and is visible as soon as printed.



opposite the printing point by the aggregate-motion mechanism, the frame supporting the typewheel is swung forward. This raises the ribbon into line with the selected type and causes the typewheel to strike forward at the platen. At the end of the printing operation, the typewheel returns to its rest position and the ribbon is lowered.

The basic design feature of the translating and printing mechanisms is the use of a stationary carriage and a moving typewheel (see Figure 4). This arrangement is superior to previous Creed teleprinters (moving carriage and fixed typewheel) as it enables the width of the machine, and hence also its weight, to be substantially reduced. It also eliminates the problem of moving and smoothly arresting the mass of the paper carriage, makes the printed copy easy to read, and allows for varying the paper storing and feeding arrangements, particularly where external pre-printed sprocket-feed business stationery manifolds are employed.

The aggregate-motion mechanism and the typewheel embody a number of important design features that combine to reduce to a minimum the energy that has to be dissipated in bringing the typewheel to rest. Consequently, it has been possible to make the components associated with the typewheel smaller and to increase their life considerably. The direct method of printing that has been employed, for example, not only dispenses with the need for a typehammer and sliding types, but permits a much-smaller typewheel. Again, the aggregate-motion mechanism moves the typewheel smoothly to the printing position by the shortest route and more slowly than on previous machines.

The keyboard is of the motorized kind: that is, the power to move the combination bars is provided by the machine itself and not by the key-depressions. In this respect, the keyboard is similar to an existing Creed model used on telex teleprinters, but the touch has been improved by making it possible to depress any key well before the end of the cycle of operations initiated by depression of the preceding key. This allows the operator to type at an irregular speed about the cadence speed of the machine.

The most-novel feature of the keyboard, makes it much simpler than other telep keyboards, is the absence of a separate key transmitter. When a key is depressed, th for the key is set up on 5 combinatio These mechanically select the code on th in the translator unit. The transmitter co which are also mounted on the translato

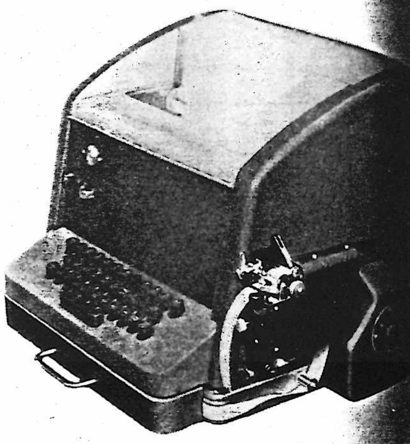


Figure 5—Close-up of reperforating attachme fits on the right-hand side of the machine and incoming messages in standard five-channel tap taneously with normal page printing. It also ena machine to be used as a normal keyboard perfor the origination of punched tape with coincident page copy.

read off the combination from the selecte and transmit it to line. This mode of op avoids the need for a separate keyboar shaft, clutch, and gears, thereby ach simplification in design and reduction in of parts. Also, since the combination is r by the aggregate-motion mechanism and p at the same time as it is transmitted to li not, as on previous machines, after a time the response of the machine to the ope touch approximates that of a typewriter.

A consequence of using the translator ca for controlling keyboard transmission is th transmission is basically 6½ units in lengt milliseconds at 50 bauds. It is, therefore, sary to insert the missing unit, and this by employing a spring-controlled time-delay mechanism that extends the stop signal from 1 unit to 1½ units in length.

3. *Further Developments*

As previously mentioned, the model-Seventy-five teleprinter, while being a complete and self-contained machine, is also the basic unit of an integrated line of equipment that is being developed by the addition of special attachments or by modifying the teleprinter in various ways.

In designing the teleprinter, a great deal of attention was given to ensuring that this projected range of auxiliary machines could be developed from the parent machine with a minimum number of modifications. Such integrated design methods result in considerable advantages to customer and manufacturer alike. Smaller stocks of spare parts can be held by customers having more than one of the related types of machine, while overall development time is reduced and manufacturing economies are made possible.

The first of the special attachments—a reperforating unit—which has recently gone into production, is illustrated in Figure 5. This fits on the right-hand side of the teleprinter, its code bars being set by the same operating levers that control the aggregate-motion-mechanism control rods. The attachment provides a perforated record of both transmitted and received signals. The punched tape, which is supplied from a drawer under the main base, issues from the attachment towards the front of the machine. Its compactness may be judged from the fact that it adds less than 3 inches (7·6 centimetres) to the width of the machine and about 1 inch (2·5 centimetres) to its height.

The other special attachments and derivative machines will include an automatic tape transmitter, a printing reperforator, and a tape teleprinter. These are, at present, being actively developed.

The mechanical local-record feature of the teleprinter enables it to be adapted easily to a variety of applications in the field of integrated data processing. One such application, which is now under development, is a teleprinter operating from 5-wire parallel input. This has been derived from the standard machine by replacing the keyboard with five electromagnets, which mechanically set the selecting pins through existing mechanisms in response to the parallel input signals.

The model Seventy-five teleprinter itself is now in full-scale production (Figure 6). Prior to this, preproduction models of the machine were given extensive trials by a number of telegraph administrations and other large organisations. These trials were most successful—how successful can be judged from the fact that orders for several thousand machines have already been received.

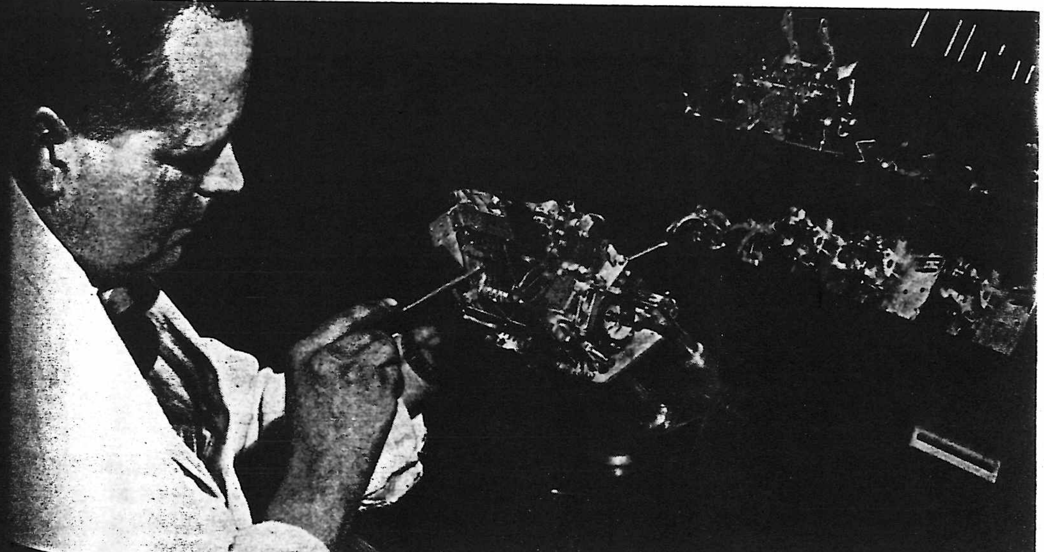


Figure 6—Assembling translator units for the model Seventy-five teleprinter.

Etching of Oliver Joseph Lodge

Oliver Joseph Lodge (1851–1940), British scientist, is depicted in the latest of the series of etchings published by the International Telecommunications Union.

Although originally planning on a business career, Lodge's interest in science led him to enter University College, London, in 1872. Starting his work as a teacher three years later, he succeeded in 1881 to the chair of physics at University College, Liverpool. In 1900, he was appointed the first principal of Birmingham University, where he continued until his retirement from academic life in 1919.

He did original work on lightning, the source of electromotive force in voltaic cells, electrolysis, ionic velocity, electromagnetic waves and wireless telegraphy, motion of the ether, and the use of electricity to disperse smoke and fog. His experiments at Liverpool with the coherer and the phenomenon of tuning, which he fully described in a patent of 1897, were important advances in radio communication. By his writings, Lodge did much to familiarize the lay public with the scientific views of his time. He also published exten on psychical research.

Lodge was elected a Fellow of the Royal ety in 1887 and received knighthood in He was a president of the Physical Society British Association, the Radio Society, and Röntgen Society The In tion of Electrical Engineers ferred on him successively vice-presidency, honorary bership, and Faraday Med

The etching of Lodge 24th in the series tha started in 1935. On a good of paper measuring 9 inches (23 by 17 centimete cluding margins, these et are available at 3 Swiss each from Secrétariat de l'Union internationale Télécommunications, Wilson, 52, rue des Genève. Suisse. The entire series is com of etchings of Ampère, Armstrong, Bell, Erlang, Faraday, Ferrié, Fresnel, and Weber. Heaviside, Hertz, Hughes, Kirchhoff. Lodge, Lorentz, Marconi, Ma Morse, Popov, Pupin, Rayleigh, Siemens Tesla.

High-Frequency Radio Receiver RX.5C

By L. J. HEATON-ARMSTRONG and J. D. HOLLAND

Standard Telephones and Cables Limited; London, England

DESIGNED primarily for the reception of frequency-shift telegraphy in the band from 2 to 30 megacycles per second, the *RX.5C* radio receiver incorporates a number of important features to improve operation under difficult receiving conditions.

The reception of telegraph signals in the high-frequency band is adversely affected by noise, interference, and fading, and it is usual to employ high-gain directive antennae, frequency-shift telegraphy, and dual space diversity to combat these effects.

It has been pointed out[1] that if selective fading is present, the use of the conventional limiter and discriminator for frequency-shift reception does not utilize all of the available information in the signal; some of this is destroyed in the limiter. If, for instance, selective fading is causing strong mark and very-weak space signals to be produced at the receiver input, the signal from the discriminator will consist of good mark signals followed by bursts of noise or the absence of signals, depending on the receiver gain adjustment. If, however, the signals could be observed on an oscillograph it would be immediately apparent that the noisy or missing elements should be space signals.

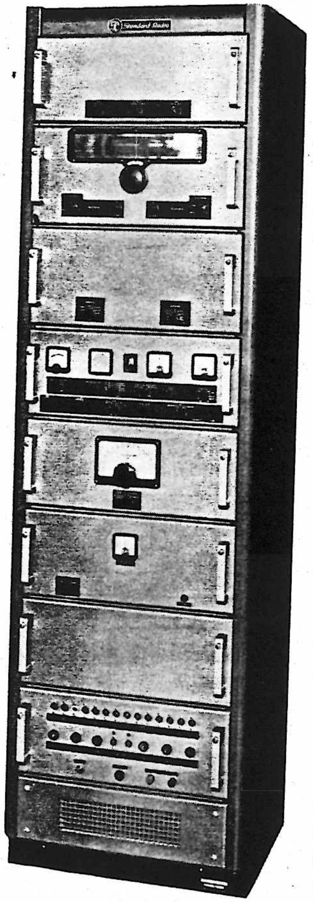


Figure 1—*RX.5C* equipment for frequency-shift telegraphy reception on the high-frequency radio band.

The receiver described here uses a technique that gives satisfactory reception provided a good mark or a good space signal alone is being received. This is equivalent to adding dual frequency diversity, and gives a decisive improvement when selective fading is present.

Another feature is the use of ratio squaring[2] for combining the dual space-diversity signals. Compared to the more-usual method of switching to the strongest signal, ratio squaring gives a worthwhile improvement in signal-to-noise ratio and eliminates trouble due to switching transients.

Another useful feature is the derivation of the voltage for automatic frequency control from both mark and space signals, so that good automatic frequency control is obtained as long as either signal is present. This mode of operation gives less residual mistuning than the use of one signal alone.

1. *Equipment*

Figure 1 is a photograph of the *RX.5C* receiver, which is housed in a cabinet 78 inches (198 centimetres) high, 23 inches (58 centimetres) wide, and 21 inches (53 centimetres) deep. The equipment is mounted on eight withdrawable trays and is split into small chassis units for ease of servicing and maintenance.

---

[1] J. W. Allnatt, E. D. J. Jones, and H. B. Law, "Frequency Diversity in the Reception of Selectively Fading Binary Frequency-Modulated Signals," *Journal of the Institution of Electrical Engineers*, Part B, volume 104, pages 98–110; March, 1957.

[2] L. R. Kahn, "Ratio Squarer," *Proceedings of the IRE*, volume 42, page 1704 (Correspondence); November, 1954.

was necessary because present traffic requirements for Trans-Canada telephone circuits have exceeded the original forecast.

Experience gained to date shows that stacking of television and telephone circuits is entirely feasible and is economically attractive where the telephone circuit requirements are limited and system lengths range from about 200 to 1,000 miles. The application of stacking has proved a worth-while contribution economically, and the transmission objectives which were defined as acceptable in the initial planning stages have been met.

References

1. THE TD-2 MICROWAVE RADIO RELAY SYSTEM, A. A. Roetken, K. D. Smith, R. W. Friis. *Bell System Technical Journal*, New York, [illegible] 30, pt. II, Oct. 1951, pp. 1041–78.

2. CARRIER TERMINALS FOR THE L3 [illegible] D. B. Penick. *Bell Laboratories Record*, [illegible] N. Y., vol. 33, no. 2, Feb. 1955, pp. 72–77.

3. L3 COAXIAL SYSTEM—TELEVISION T[illegible] J. W. Reike, R. S. Graham. *AIEE Tr*[illegible] vol. 72, pt. I, Nov. 1953, pp. 541–51.

4. SWITCHING AT TV OPERATING [illegible] A. L. Stillwell, A. D. Fowler. *Bell L*[illegible] *Record*, vol. 34, no. 10, Oct. 1956, p. 366.

5. WIRE LINE ENTRANCE LINKS FOR TD-[illegible] Beyer. *Ibid.*, vol. 32, no. 3, Mar. 1954, pp. [illegible]

Error-Checking for 5-Channel Telegraphic Tape

R. A. BARBEAU
MEMBER AIEE

THE USES and the advantages of 5-channel paper tape are well recognized in the business world. Paper tape has been used for many years for message sending. Its use in telegraphic communication equipment serves as the life line of communications for many businesses. In recent years the decentralization of factory equipment and the general growth of businesses have made the transmission of accounting or scientific data between cities increasingly important. The rapidly expanding use of commercial computers and business machines will result in a further increase in the volume of data that must be communicated between cities. As more and more data are communicated, one becomes aware that a major improvement is required to refine the system, especially with regard to error handling. There is a practical limit on the improvements that may be made in transmission facilities to decrease the error rate. However, these improvements, plus a system of error detecting, could provide a highly efficient system that would be able to handle large volumes of data with a minimum amount of error-handling techniques.

This paper will describe an error detection system that is compatible with present-day telegraphic communication facilities, including automatic switching centers. It will describe a complete, practical system that has been tested and has proved to be highly efficient. The paper will discuss in detail the systems problems and engineering problems, and how they were approached.

Systems Problems

Systems used in the past have made use of group total control for the detection of errors. Information prepared at a point of origin, such as invoice writing, may be contained in a roll of tape 100 to 200 feet in length. At the point of origin a group total is compiled for the important items that should be checked, such as quantity of items or amounts of money. As shown in Fig. 1, this information is transmitted to a distant reception point, the conversion to other means of storage takes place, and it is followed by an attempt to balance to the control figures. Should the control totals fail to balance, there is at present no easy means of indicating or pinpointing where the error might have occurred in this group of records. It can, in fact, necessitate the reprocessing of a new tape at the point of origin for the full amount of information that did not balance. This is not only time-consuming and costly, but inefficient, because both correct and erroneous information is reprocessed.

It is known that there are five areas in which errors may occur, as follows:

1. The preparation of paper tape, especially when done by a manual method.
2. Reading of the paper tape for transmission.
3. Actual transmission of data over [illegible] lines or other media.
4. Reperforation of paper tape at a distant point of reception.
5. Reading of tape for conversion.

To have a practical system of error detection, any rearrangement of [illegible] must not have an adverse effect on communication equipment now in use, such as page printers or other subscriber [illegible] In fact, equipment already available in the field must be utilized. Rearrangement of codes to include checking must [illegible] make unintelligible or otherwise affect information printed on page printers. Basic telegraphic code arrangement [illegible] must not be violated.

Any checking code must not interfere with switching centers now in existence. As an example, a figures-shift *H* character must not be used for record checking because this code has its primary use [illegible] switching center control.

The present economy in transmission time must be maintained, which [illegible] that any added characters for checking must be kept to a minimum.

In a complete system that is to [illegible] broad usage, any proposed device capable of producing checkable tape must [illegible] electrical inputs from other business machines.

The problem of operator error [illegible] manually keyed machine is probably [illegible] of the most difficult items to handle in the over-all system of error control. Many operator errors may be eliminated by a simplified error correction procedure and by a reduction in the number of operational decisions required of the [illegible] punch operator.

Design Goals

After the systems problems mentioned had been reviewed, it was realized that design efforts should be concentrated in the areas of tape preparation and [illegible]

Paper 58-2, recommended by the AIEE Data Communication Committee and approved by the AIEE Technical Operations Department for presentation at the AIEE Winter General Meeting, New York, N. Y., February 2–7, 1958. Manuscript submitted July 12, 1957; made available for printing October 8, 1957.

R. A. BARBEAU is with the International Business Machines Corporation, Poughkeepsie, N. Y.

The author wishes to express his appreciation to D. K. Close and P. M. Young for their helpful suggestions and criticisms in connection with the preparation of this paper.

Communications + Electronics May 1958 #36

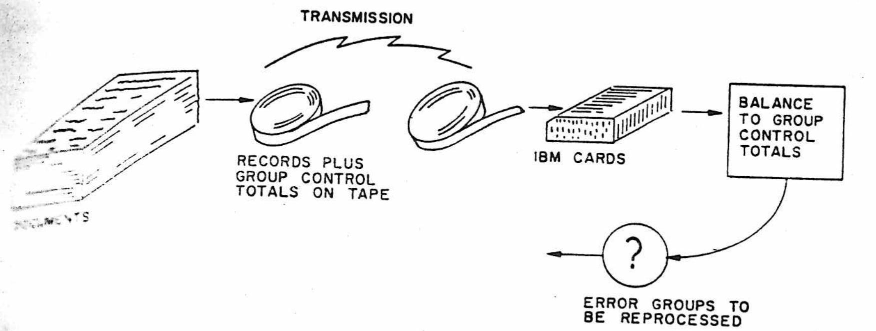

TRANSMISSION
DOCUMENTS
RECORDS PLUS GROUP CONTROL TOTALS ON TAPE
IBM CARDS
BALANCE TO GROUP CONTROL TOTALS
?
ERROR GROUPS TO BE REPROCESSED

Fig. 1 Data-flow chart

conversion. It also became evident that, to detect error at the converter, some type of "piggy-back" check should accompany the information in the form of a tape code

The major goal was to design and supply an entire workable check system from start to finish, from the preparation of tape to the finished accounting medium, such as cards, for example.

Another important design goal was aimed at the operator preparing the punched tape. It was recognized that the operator plays a very important part in the accuracy of the over-all system. Consequently, any changes that would simplify the operator's work in the proper manipulation of keys, the reduction of key strokes, or other improved operating procedures would aid in producing a more accurate and better tape from an over-all systems standpoint.

Another objective was to design an error deletion routine that would be fast, automatic, and reliable, possibly controlled by a single keystroke. This would serve to relieve the operator of multiple decisions during the tape preparation.

The design of the system should keep total costs in line with the results gained. The system must contain the desired accuracy without imposing a hardship on the required transmission time, and it must not require changes in present telegraphic transmitting facilities.

The code arrangement, if changed to include checking, should not affect any of the existing systems in telegraphic communications. Particularly, it should not affect functions attributed to automatic switching centers. Also, such a code arrangement must be adopted that can make use of all telegraphic facilities.

Preparation of Checkable Tape

A variable length block of information was decided upon as a unit record of information. A unit record of this nature would fit a telegraph page printer, a typewriter, or a tabulating card. It was found that a unit record which corresponded to a line on a printed page was compatible for both error deletion and error detection.

Error Deletion

If deletion is to be automatic, an identification means must be used to signify the beginning of a record and again to indicate the point of error. The machine preparing the tape is basically a punching mechanism that does not contain a multiple-channel tape-reading station. It was decided to use a notch on the top edge of the tape for identification, which would allow for ease of sensing and would not require interpretation by a multiple-channel reading station. A notch is located between the last character of the preceding unit record and the first character of the new unit record. Another notch is used to identify the point of error.

If the operator recognizes a mistake during a given unit record, the depression of an error key notches the tape at the point of error. Automatic backing up of the tape to the point where the record started, as shown in Fig. 2, and automatic deletion or superimposing by the delete code will take place until a notch is read indicating the point of error. The notch in the tape will not affect the communication tape readers because its location is outside the operating area of the reading and run-out pins.

This manner of automatic error deletion fits a requirement of manual operation. No decisions are needed on the part of the operator other than to depress the error key at the time the error is recognized and to retype the line. The notch in the tape signifying a unit record is entered automatically at the beginning of every line. It is possible for the operator to make a correction for that line no matter where in the line she recognizes the error. The depression of the carriage return key may be likened to a motor bar depression at the end of an entry. The entry may be corrected at any time prior to the depression of the motor bar operation. This single decision by the operator has proved very effective and accurate in the preparation of paper tape.

Checking Code

A decision was made to arrive at a piggy-back code by counting the code bits in the record. This code was then attached as part of the record to be transmitted. The decision to count the bits in a record blended with the fact that it could be directly coded to the figures-shift digits 1 through 9 and zero. It was decided that the check digit count would be the units position of the total count of the number of holes appearing in the tape following a beginning of unit record identification and including all of the end of record identification codes. It was not felt that a 2-column check code was required, and further tests have borne out the fact that the single-column check code is sufficient.

The generation of a check code by unit record signifies that a beginning of record identification must be used, as well as an end of record identification, because the same method of checking the information at the point of conversion must take place in the system. The identification of the unit record for the operation of a checking code cannot be the notch because the notch will never appear on the reperforated tape.

Because the letters shift code (LTRS)

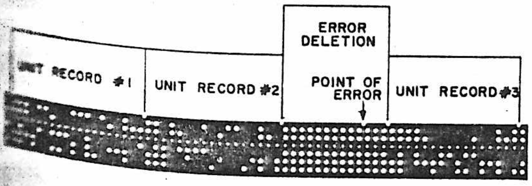
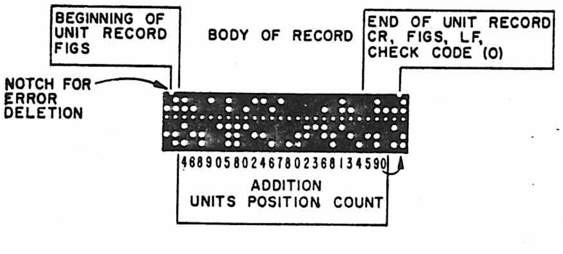


Fig. 2 (left). Error deletion

Fig. 3 (right). Unit record of tape


BEGINNING OF UNIT RECORD FIGS
BODY OF RECORD
END OF UNIT RECORD CR, FIGS, LF, CHECK CODE (0)
NOTCH FOR ERROR DELETION
46890580246780236813459 0
ADDITION UNITS POSITION COUNT

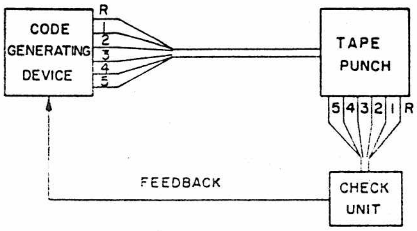


Fig. 4. Redundancy checking

is used for error deletion and as a run-in code, the figures (FIGS) shift code was selected to identify the beginning of unit record. The carriage return (CR) and line feed (LF) codes required for page printers were further expanded to provide an end of unit record identification. The check code requires a FIGS shift code, and the CR function requires a second nonprinting code. The identification of the end of unit record is then comprised of a 3-unit code group, namely, CR, FIGS, LF. The check code follows the end of unit record identification and becomes part of the unit record. A complete unit record of tape is shown in Fig. 3.

The FIGS entered between CR and LF becomes a very distinctive identification mark for the new checkable tape, as contrasted with a noncheckable tape containing CR, LF, in that sequence.

During the perforation of paper tape at the point of origin, the entry of the FIGS shift code at the beginning of a unit record conditions the machine to clear the adding counters, makes a reset zero check, and instructs the machine to add all information following this beginning of unit record identification. When the case of the next code is alphabetic, LTRS shift follows the first FIGS shift. While intelligence is being entered into the unit record of tape, addition of the holes making up the 5-channel code is in process throughout, including the end of unit record identification. The total units count of the number of holes will be recorded in a telegraphic code after the LF.

Automatic Cycles

Tape punch circuitry was designed to accept a single pulse in order to enter automatically CR, FIGS, LF, check code, and notch FIGS denoting the end and beginning of unit records. This single pulse is generated upon the depression of the typewriter carriage return key. The depression of the TAB key on the typewriter enters G, FIGS or FIGS, G, LTRS automatically into the tape. A 3-row keyboard arrangement was adopted, and the operation of the basket shift mechanism automatically enters the proper shift codes. These automatic cycles contribute to tape accuracy and ease the operator burden.

In certain cases involving page printers, it is desirable to separate the printed check code from the body of the printed record. This is accomplished by having the typewriter carriage return key depression automatically insert CR, FIGS, LF, check code, G, and notch FIGS in the tape. The G code operates the tabulating mechanism in the page printer. It is also desirable in some cases to insert LF codes following the beginning of record FIGS code. Subsequent depressions of the carriage return key insert the LF code only.

Redundancy Checking

Whenever two pieces of equipment are electrically connected, some sort of feedback is desirable to ensure the accuracy of both devices. It was decided that the code-generating device should furnish a redundancy 6-channel code, five of which were the telegraphic channels; see Fig. 4. Five telegraphic channels would be punched in the paper tape, and the sixth would operate a dummy punch. All six punches would operate circuitry to effect an odd check.

Circuits in the code-generating device are so designed that the failure of a single relay point or a single diode will cause no more than the omission of a single bit, or all bits, either of which omissions may be detected by the odd count redundancy type of check. This check takes place during the punching of every code in the tape.

Reliability Checks

In keeping with the accuracy of the system, reliability checks were added to ensure the earliest possible detection of machine malfunction and thus avoid lost time throughout the system.

As a supplement to redundancy checking, or when a code-generating device is not capable of supplying a sixth channel output, the design of the mechanism and circuitry provides a further accuracy check of the output medium and punching mechanism. On every tape-punching cycle a check is made between the input and the actual punching for each channel of tape information. Electric pulses, conditioning channels 1, 4, and 5, for example, are compared with actual holes perforated in channels 1, 4, and 5. The conversion of the decimal check digit in the counter to a telegraphic tape

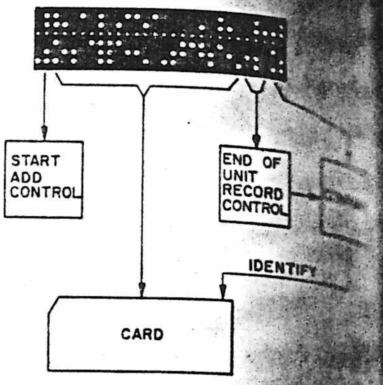


Fig. 5. Tape-to-card conver[illegible]

code is done on a 6-channel basis [illegible] ing with redundancy checking.

To assure the accuracy of the [illegible] ulated check digit, two sets of [illegible] are used. These counters are [illegible] relays whose accuracy has been [illegible] by previously mentioned methods. [illegible] comparison is made between [illegible] counters on each tape-punching [illegible]

A second notch-reading pin is [illegible] to check the accuracy of the first [illegible] reading pin. It provides assurance [illegible] good record will not be overrun [illegible] error deletion. Both notch-reading [illegible] are used to check tape movement [illegible] beginning of every record.

When malfunctions are detected [illegible] these means, a signal locks the [illegible] writer keyboard to prevent further [illegible] essing of tape. The operator may [illegible] depress the error key, deleting the [illegible] record, and attempt to retype the [illegible] Thus unit failures are caught, and [illegible] operator is prevented from [illegible] faulty tape.

Paper Tape Transmission

The tape prepared as described [illegible] now be transmitted over [illegible] communication equipment, and [illegible] perforated tape prepared at the [illegible] reception. It should be noted that [illegible] system does not require changes in [illegible] ard equipment, and any [illegible] made in service will tend only to [illegible] the system.

Error Detection During Conversion

Since the reperforated tape [illegible] checkable by unit record, its conversion [illegible] an accounting medium with error [illegible] fication by unit record may be [illegible] plished. Although a discussion of the systems aspects of a tape-to-card [illegible] verter would supply a background for the following discussion, it is felt that [illegible] conventional converter functions might be indistinguishable from the detection means incorporated in the machine. Therefore this section of the paper will deal primarily with the error detection phase of conversion.

As previously listed in the systems problem section, errors can occur during the punching of tape for transmission, during transmission, during reperforation, and during the reading of tape for conversion. In the design of a tape-converting system, a careful review must be made of the types of errors that will be encountered, and enough intelligence must be built into the machine to insure fail-safe operation. This intelligence must also include reliability checks.

The error-detecting device attached to a tape-to-card converter contains primarily a single-position decimal counter under reset and add control of the first FIGS in a record. Accumulation of the code bits takes place in much the same way as in the device which prepared the checkable tape. The piggy-back check code in the tape is compared with the total in the counter under the control of the end of unit record codes, CR, FIGS, and LF. During tape reading, the analysis of the first FIGS code of a record resets the counter, makes a zero check, and conditions the counter to add succeeding code bits. The body of the record is punched into the card simultaneously with tape reading.

When the tape converter analyzes the end-of-unit record codes, CR, FIGS, LF, the machine is instructed that the next character to be read in the tape is the check code. During the reading cycle following the LF code, a comparison is made between the accumulated decimal count in the converter and the check code appearing in the tape. When the two numbers are in agreement, the converter gives a check signal for proper card identification. During the comparing cycle, if the digit appearing in the counter is in disagreement with the code which is read in the tape, the converter gives a noncheck signal for proper error identification of the card. Because the check code appearing at the end of a unit record is strictly for accuracy checking, it is not available for conversion purposes. When the figure shift G code has been inserted between unit records, the G is not made available for conversion purposes.

UNIT RECORD CONTROL

For efficient use of this type of check digit appearing at the end of a unit record, it is necessary to have close control over defining the record limits, both the beginning and the end of unit record.

The primary use of the beginning of record FIGS code is to control subsequent adding of the code bits of the unit record during conversion. To effect a control of a record that has a mutilated beginning of unit record FIGS code, the conversion of tape codes conditions an error memory device. The device supersedes check code comparison in order to effect error identification of this record.

Previous error-detecting experience led to setting up controls using all three end-of-unit record identification codes in a given sequence. When all three end-of-unit record codes are present, CR, FIGS, LF, in that order, the comparing cycle output depends upon agreement between the counters and the check code in the tape; see Fig. 5. If one of the three end-of-unit record codes has been mutilated, the error memory device is controlled and will cause the noncheck signal during the comparing cycle. Examples of these code mutilations follow:

Space, FIGS, LF
CR, V, LF
CR, FIGS, CR

The converter will acknowledge any two of the three end-of-unit record codes in correct sequence and cause comparing on the correct cycle. This comparing cycle control is not effective when an LF code follows the beginning of record FIGS.

If CR, LF appear to the converter in this order, they cause automatic comparing on the next cycle with an automatic noncheck signal output. This comparison serves two purposes:

1. Indication that noncheckable tape is being processed.
2. Prevention of switching directive codes from being processed as a correct record.

When the CR code is sensed not in its proper position, the error memory device will be operative; for example, if a channel 3 space code has been shifted to a channel 4 CR code in the body of the unit record.

Prior to testing this error detecting system, it was recognized that certain types of compensating errors (such as the dropping of one bit on one code and the adding of another bit on another code within a unit record) would not be detected solely by the check code appearing at the end of the unit record. It was also realized that certain types of transmission failures might result in the dropping or adding of ten bits of information. During the testing of this system, it was found that the compensating errors within a unit record were very infrequent, and the occurrence of an error of ten was extremely rare. The logic of the converter, prior to the application of error detection, could already take care of some of these compensating errors or errors of ten, and these checks were available at no extra expense to the system.

Conclusions

Numerous papers have been given on the subject of error detection, mentioning a check code. However, it is felt that for a system to be practical it should contain close limit control on the checking block. These controls make it possible to use a variable length record with a high efficiency of error detection. The system described here is aimed primarily at compatibility with the existing commercial communication facilities.

Conventional error handling in accounting systems is time-consuming and costly, because error detection is normally done by control groups at a point in accounting later than tape conversion. The checking system described in this paper allows for error detection by unit record during the conversion process. Reprocessing or retransmission of only the erroneous information saves considerable time. This system proves to be an accurate, practical checking means for paper tape.

1958 Western Joint

Communication Between Computers

WILLIAM S. KNOWLES
NONMEMBER AIEE

IRVING L. WIESELMAN
NONMEMBER AIEE

RAYMOND STUART-WILLIAMS
NONMEMBER AIEE

THE USE of electronic computers to solve large-scale data processing problems has created a number of serious communication problems. Not the least of these problems is the necessity that machines be capable of communicating with each other. To achieve complete communication between machines requires that three conditions be satisfied. First there must be compatibility of the medium used for transference. Second, there must be understanding of the coded language used; and third, there must be the ability to interpret the coded language as structured by the program.

Unfortunately, present machines generate and receive information in different languages and structures. Even identical machines can and do change their language by program means. These differences are not attributable to the ignorance or to the whims of machine designers but instead are due to the operational requirements for which the machines have been designed. It seems clear that for some time, future computers will necessarily follow this pattern of incompatibility.

There are two basic methods which may be used to resolve the communication problem. The first is to legislate that a common medium, code, and language be used, and the second, is to employ other machines to solve some or all of the compatibility problems involved in machine communication. Legislation has the disadvantage that it would involve a tremendous loss of investment, in terms of existing programs, and would also stifle the future development of computers. The second approach is to use machines which may be programmed to interpret the languages of different machines. This does not introduce these disadvantages and, therefore, has been adopted throughout the industry as the preferred approach.

Communication through a transfer medium is almost always employed. Some possibilities for this are printed forms, paper tape, punched cards, and magnetic tape. Magnetic tape, because of its high packing density and high rates of access, is becoming the accepted medium for communication. The use of a transfer medium always introduces both physical and linguistic differences. Two classes of machine will be defined to handle these differences. The first class is termed "media translators." These are used to overcome the physical differences. They transfer information from one medium to another without radical changes of the language employed. The second class is termed "data converters," and these are used to overcome both physical and linguistic differences. They effect both transference and translation of information between media.

In the future a third class of machine will be introduced. This is simply an adapter attached to the existing data processing computer and represents the minimal form of media translator. The media translator will continue to exist as a special-purpose instrument. The data converter will evolve into a machine which uses stored program computing techniques to solve the linguistic problem.

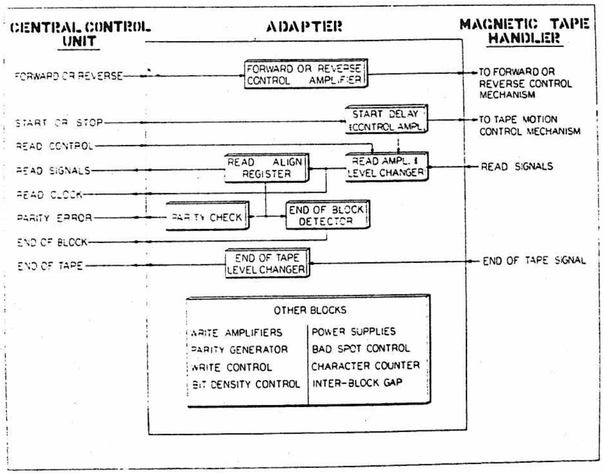


Fig. 1. Typical magnetic tape adapter block diagram

The Compatibility Problem

Historically, computers were designed as either scientific or data processing machines. Each machine was considered from a system point of view, as a separate entity. As the number of data processing machines in everyday use have increased, the results obtained from one machine are used as the input data to a second machine, which is often geographically remote from the first, and which is frequently of different type and manufacture. The direct approach used to solve this problem is to employ either printed paper, punched tape, or punched cards as the transfer medium. Thus, either written human language, teletype code, or punched card code are used.

The use of magnetic tape, as a means of mass data storage, has become more common and this is also being widely adopted as a high-speed transfer medium. It is apparent that future machines will need an increasing ability to intercommunicate. This fact has been recognized to a small degree by the manufacturers who are building new machines with capacious buffered magnetic-tape storage systems.

However, it seems clear that near term machines of the future will not utilize a common language or common magnetic tape equipment. Indeed, it is perhaps not in the best interest of machine data processing progress to permit stasis of this type. Therefore, it may be safely concluded that problems in communication will exist to an increasing degree in the foreseeable future.

In communicating from one machine to another there are both physical and linguistic differences which together create what is generally called "the compatibility problem." These problems can be defined as follows:

PHYSICAL-MECHANICAL

These are the differences: In the tape material; the width of tape; the number of heads; and the head spacing. Further problems arise due to the fact that the rate of tape motion, the acceleration time, and deceleration time vary in the different types of transport. These physi-

---

WILLIAM S. KNOWLES, IRVING L. WIESELMAN, and RAYMOND STUART-WILLIAMS are with Telemeter Magnetics, Inc., West Los Angeles, Calif.

cal-mechanical differences are so gross that there is very little possibility of devising a tape transport which can accept all types of magnetic tape.

PHYSICAL-ELECTRICAL

These are the differences: The system of recording employed, such as return to zero or nonreturn to zero; the information packing density and the rate of tape motion which define the character rate. Certain systems have means whereby the condition of the tape can be checked and bad spot areas rejected. Obviously, this operation involves special circuits and delays peculiar to each type of transport.

The foregoing difficulties are resolvable by the use of relatively unintelligent electronic equipment. In essence, units defined as "adapters" are used to convert the signals in and out of a particular type of tape transport to a standard from, acceptable to the computer or to a media translator or to a data converter.

LINGUISTIC-FORMAL

These are the differences found between tapes produced from different data processing machines utilizing various formats. These differences exist even when tapes are produced on the same tape transport but fed from different machines. Typical differences are found in:

1. Characters: 4-, 5-, 6-, 7-, or 8-bit codes are employed to represent characters.
2. Words: Fixed or variable word length is employed.
3. Blocks: Fixed or variable block length is employed.
4. Sign: Special bits or special characters are employed, either at the beginning or end of words.
5. Checking: Odd or even lateral parity, longitudinal parity, or count of characters for fixed block length.

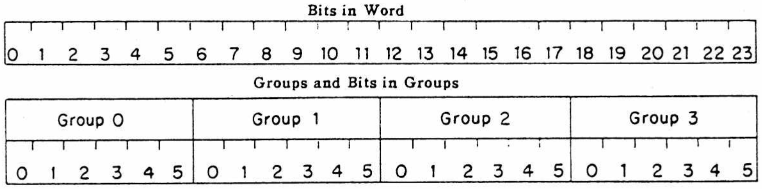
These problems can be solved in most cases by the use of straightforward computing-type circuitry but without requiring programmed operations to perform conversions.

LINGUISTIC-VARIABLE

The major linguistic problem associated with data conversion is the variation from program to program. Linguistic formal differences can be variable in some circumstances. For example, a binary machine may be programmed to produce a coded output. This code is entirely under the control and at the convenience of the programmer. Under these circumstances code problems cannot be defined as a formal difference. As another

Table I. Generalized Central Control Unit Command List

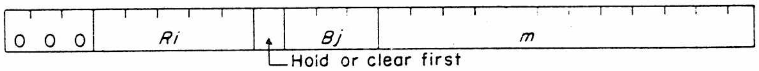
Word Structure



Command Codes, Structure and Function

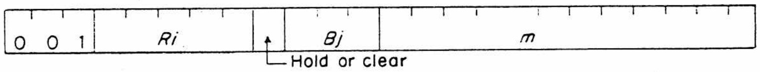
Command Code — **Command Structure** — **Execution Time** (Add 3 microsecond (μsec) for B Mod.)

0_i



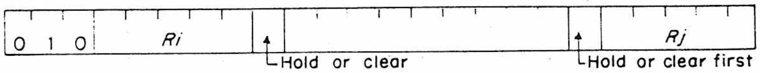
$(m') \rightarrow R_i$* — 12–15 μsec

1_i



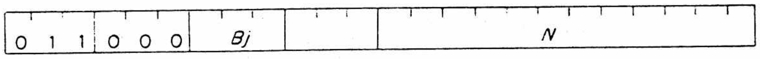
$(R_i) \rightarrow m'$ — 12 μsec

2_i



$(R_i) \rightarrow R_j$ — 12–15 μsec

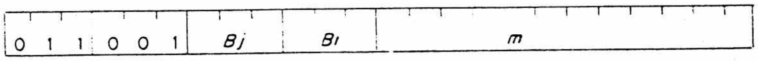
30



Add N to B_j — 16 μsec

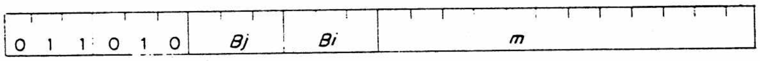
* R_i Registers are listed in Table II with name and number

31



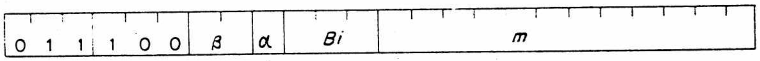
Jump to m' if $B_j = 0$ — 16 μsec

32



Add m' to B_i — 20 μsec

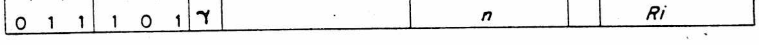
34



$\gtreqless$ jump to m'

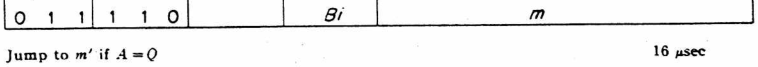
$\alpha = 0$ Refers to A
$\alpha = 1$ Refers to Q
$\beta = 00$, jump if zero — 16 μsec
$\beta = 01$, jump if > zero — 16 μsec
$\beta = 10$, jump if < zero — 16 μsec
$\beta = 11$, unconditional jump — 7 μsec

35



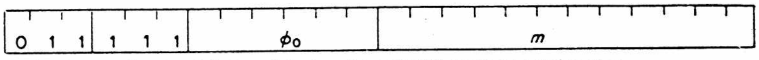
Shift R_i right n places $\gamma = 1$ — 12 μsec plus
Shift R_i left n places $\gamma = 0$ — 6 μsec (number of bits in n)

36



Jump to m' if $A = Q$ — 16 μsec

37



MK-0 Program input: Selects unit in ϕ_0 and inputs 512-m words, starting at m
MK-1 Program input: Selects unit in ϕ_0 and input K_I words, starting at m

Table I. (Continued)

40

1 0 0 0 0 0	ϵ	α	Bi	m	δ

Select group δ of m' and place in group ϵ of $A(\alpha = 0)$ $[Q(\alpha=1)]$, replacing ϵ 30 μsec

41

1 0 0 0 0 1	ϵ	α	Bi	m	δ

Select group ϵ of $A(\alpha=0)$ $[Q(\alpha=1)]$ and place in group δ of m', replacing δ 33 μsec

42

1 0 0 0 1 0		Bi	m

Stuff address of NIA register plus one in m' 16 μsec

43

1 0 0 0 1 1	ν	α	Bi	m

Form logical product of $E\nu(\nu=00, 01, 10, 11)$ with (m') and then logically sum with $A(\alpha=0$ $[Q(\alpha=1)]$ 30 μsec

44

1 0 0 1 0 0	λ	α	Bi	m

Add (m') to $A(\alpha=0)$ $[Q(\alpha=1)]$, Sum in $A[Q]$
$\lambda=00$, full add 44–70 μsec
$\lambda=10$, add groups 2 and 3 20–28 μsec
$\lambda=11$, add group 3 20–28 μsec

45

1 0 0 1 0 1	λ	α	Bi	m

Subtract (m') from $A(\alpha=0)$ $[Q(\alpha=1)]$, Difference in $A[Q]$
$\lambda=00$, full subtract 50–78 μsec
$\lambda=10$, subtract groups 2 and 3 26–34 μsec
$\lambda=11$, subtract group 3 26–34 μsec

46

1 0 0 1 1 0	λ		Bi	m

Multiply A by m', most significant portion of product in A, least significant portion of product in Q
$\lambda=00$, full multiply 0.6–1.8 millisec
$\lambda=10$, multiply groups 2 and 3 0.3–0.8 millisec
$\lambda=11$, multiply group 3 0.14–0.3 millisec

47

1 0 0 1 1 1	λ		Bi	m

Divide A by m', quotient in Q, remainder in A
$\lambda=00$, full divide 1.7–3.2 millisec
$\lambda=10$, divide groups 2 and 3 0.6–1.1 millisec
$\lambda=11$, divide group 3 0.25–0.4 millisec

$5i$

1 0 1	σ	ν	Bi	m

Jump to m' if flip-flop $F\sigma\nu$ is set 8 μsec
$\sigma=0\ 0$ to $1\ 1$; $\nu=0\ 0\ 0\ 0$ to $1\ 1\ 1\ 1$

6σ

1 1 0	0	σ	W		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Set $(W=1)$ [Reset $(W=0)$] Selected flip-flops $F_{\sigma\nu}$ 8 μsec
$\sigma=0\ 0$ to $1\ 1$; ν represented by bit position

64

1 1 0	1 0 0	W		σ	ν

Set $(W=1)$ [Reset $(W=0)$] selected flip-flop $F_{\sigma\nu}$ 8 μsec
$\sigma=0\ 0$ to $1\ 1$; $\nu=0\ 0\ 0\ 0$ to $1\ 1\ 1\ 1$

70

1 1 1	0 0 0	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4

ϕ_0 Selects one of 63 units for flow path 0
ϕ_1 Selects one of 63 units for flow path 1
ϕ_2 Selects buffer 1 or 2 for flow path 2
ϕ_3 Selects buffer 1, 2 or R register for flow path 3 8 μsec
ϕ_4 Selects buffer 1, 2 or R register for flow path 4

$74+\theta$

1 1 1	1 0 θ	$\mu 0$	$\mu 1$	$\mu 2$	$\mu 3$

Input-Output mode conditioning
$\theta=0$, flow path 0 selected
$\theta=1$, flow path 1 selected
$\mu 0$ selects fixed mode of operation for selected unit
$\mu 1$ selects variable modes of operation which may be changed conditional on bits selected in μ_2 and μ_3
$\mu 2$ selects which bits of μ_1 will be changed 8 μsec
$\mu 3$ selects the conditions for change

77 Stop

example, it is possible to define formal rules whereby the word length or the block length is changed during the process of a data conversion. However, in many programs the word may, in fact, have an implied sign which may be defined by its position in a block and, therefore, change of either a word length or a block length can destroy this implicit meaning so that formal methods are not usable in this conversion.

The problem of translating computer languages is similar to that of translating spoken languages. Each spoken language has its own alphabet, words and grammatical structure. The meaning of a word is a function of the grammatical indicators associated with its common usage, and the surrounding context according to the rules of syntax. The words in the output language are chosen by its structure so as to represent the same meaning as the input. The difficulty in translating spoken languages is that the language structure is so complex that it is difficult to define the rules of translation. However, in the case of translating computer languages no matter how complex the structure is, and how it varies from sequence to sequence, the rules which governed the original program may be stated. Translation may be accomplished provided the computer is capable of performing the rules of operation, and this, in general, can only be done by a fully programmed controlled conversion.

Solution to the Problem

The general problem of machine compatibility is extremely complex. However, solutions are possible using data-processing computers. If adapters or media translators are employed, then either the generating or receiving computer is programmed to perform the actual translation of information. If, on the other hand, a data converter is used, then this must be defined as a special-purpose off-line computer which is used to perform the translation.

Adapters

An adapter is a device which allows a computer to communicate with input or output equipment. Fig. 1 shows some of the functions which must be performed in a typical magnetic tape adapter so that information on the tape may be read or written. Information signals are received and transmitted in a form compatible with the computer. Control signals are interpreted and cause prescribed actions in the tape handler. Signals are also generated in the adapter which indicate the state of the tape or control information such as parity errors, end of block, and end of tape. The timing for writing information is generated as well as logic and delays for special functions such as tape bad spot controls.

All data-processing computer designers face the problem of building adapters for their own input or output devices. The designer of data conversion computers must build adapters so that alien input-output units can communicate information to the conversion equipment. The design problems differ only in that the data processing designer may simplify the adapter by considering the total system.

It is possible for machines of newer design which have adequate buffering and sufficient input-output tape positions to add adapters so that alien tapes could be processed. The adapters can also be employed for producing alien tapes for use on other machines.

Media Translators

A media translator consists of two or more adapters, plus a central control unit which performs those conversions necessary to introduce compatibility to the physical and formal linguistic level, but not at the linguistic variable level. Fig. 2 shows a media translator built by Telemeter Magnetics, Inc. to perform translations between Remington Rand Univac *1103A* and International Business Machines Corporation (IBM) *704* magnetic tapes in either direction. See functions of Remington Rand Univac *1103A* and IBM Media Translator. In addition to solving the physical compatibility problem, it checks for parity and count errors; it can be used for editing by translating selected blocks; and it solves a formal linguistic problem by inputting continuous data and outputting data in block form.

The central control unit uses a standard Telemeter Magnetics, Inc. *1092-BQ-8* magnetic core buffer to store information. It also has presettable decade counters and transistorized logical and control circuits.

The reasons for existence of media translators are quite logical. When dealing with alien tapes, adapters of some form are required in any case. It is often desired to produce a copy of an original tape for transmission which wastes time for most modern machines. An off-line machine which performs a copying operation is almost as complex as a complete media translator. Media translators are also used to solve compatibility problems by translating information between dissimilar media such as between paper tape, magnetic tape, punched cards, or from unformatted real-time data. For these reasons, and other economic considerations, the media translator is a logical and useful development. It is, however, severely restricted in application as it is incapable of performing any translations other than those defined as formal linguistic differences.

Data Converter

The problems involved in data translation can be solved by using stored program computing methods. The computation required is trivial and repetitive. It is an accepted principle that under these conditions a special-purpose computer is always faster and less expensive than a general-purpose machine. It is on this basis that the data converter exists as a class of equipment. Data converters are fundamentally media translators, to which have been added a special-purpose computer to perform the translation functions. Present data converters are rudimentary in that they are sequence-controlled machines in which the sequence is controlled only by means of a plugboard. Machines of this type are not sufficiently flexible to perform many significant translation operations. They can, however, perform quite complex changes in word and block structure. They can also make simple code changes and can perform some editing functions. The methods of error detection that are employed are quite sophisticated and it is possible to program them to make simple error corrections.

A data converter, built by Telemeter Magnetics, Inc., for the Air Force Missile Test Center (AFMTC) is shown in Fig. 3. It was built so that missile test data produced at AFMTC could be made compatible with formats of various commercial computers in operation at participating missile development organizations in scattered locations.

The present input-output equipment allows for conversion between the AFMTC, Florida Automatic Computer (FLAC) computer paper and magnetic tapes, the IBM *704* and IBM *650* magnetic tapes, and the Remington Rand Univac *1103A* paper tape. (See list of conversions and functions of AFMTC data converter) It has plug-boards and switches which are used to control editing and conversion functions such as selecting given blocks and given words within blocks for conversion, translating from variable word length to fixed word length, changing the position of code for

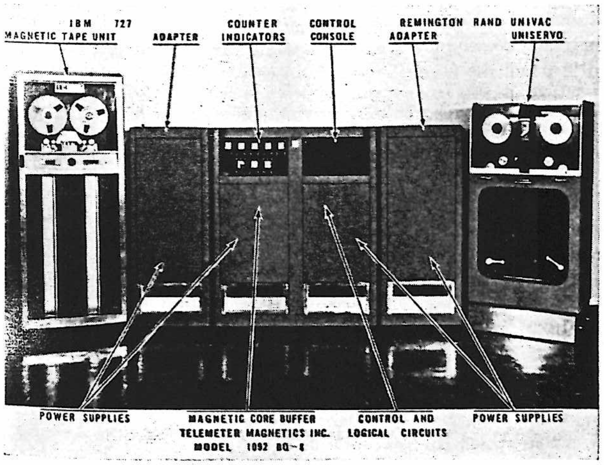


Fig. 2. Remington Rand Univac 1103A-IBM 704 media translator

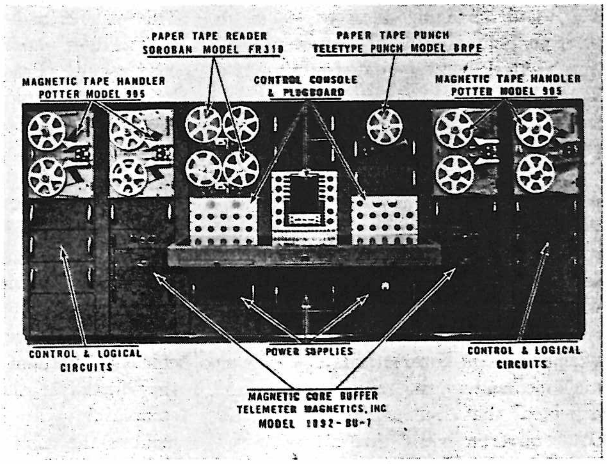


Fig. 3. AFMTC data converter

Table II. Generalized Central Control Unit Addressable Registers, Bit Transfers, Function, and Type

Register Name	Register No.	Write	Read	Function	Type
B_k	$0k$ $(k=1,2,3)$	$R_j \to B_{kj}$ $(j=2,3)$	$B_{kj} \to R_j$ $(j=2,3)$	index register	core storage
B_m	$1k$ $(k=0,1,2,3)$ $(m=4,5,6,7)$	$R_j \to B_{mj}$ $(j=2,3)$	$B_{mj} \to R_j$ $(j=2,3)$	index register	core storage
A	20	$R_j \to A_j$ $(j=0,1,2,3)$	$A_j \to R_j$ $(j=0,1,2,3)$	accumulator	core storage
Q	21	$R_j \to Q_j$ $(j=0,1,2,3)$	$Q_j \to R_j$ $(j=0,1,2,3)$	Quotient, LSP of product, or accumulator	core storage
N	22	$R_j \to \bar{N}_j$ $(j=0,1,2,3)$	$\bar{N}_j \to R_j$ $(j=0,1,2,3)$	complementor	core storage
L	23	$R_j \to L_j$ $(j=0,1,2,3)$	$L_j \to R_j$ $(j=0,1,2,3)$	stores location of initial address when filling or emptying internal memory	core storage
E_k	$3k$ $(k=0,1,2,3)$	$R_j \to E_{kj}$ $(j=0,1,2,3)$	$E_{kj} \to R_j$ $(j=0,1,2,3)$	extractors	core storage
W_L	40	$R_j \to W_{Lj}$ $(j=0,1)$	$W_{Lj} \to R_j$ $(j=0,1)$	stores left half of word	core storage
W_{LS}	41	$R_j \to W_{LSj-2}$ $(j=2,3)$	$W_{LSq} \to R_q$ $(q=0,1)$	shifts right half of word into left half of word	core storage
W_R	42	$R_j \to W_{Rj}$ $(j=2,3)$	$W_{Rj} \to R_j$ $(j=2,3)$	stores right half of word	core storage
W_{RS}	43	$R_j \to W_{RSj+2}$ $(j=0,1)$	$W_{RSq} \to R_q$ $(q=2,3)$	shifts left half of word into right half of word	core storage
F_k	$5k$	$R^i \to F_k^i$ $(i=0,1,..16)$	$F_k^i \to R^i$ $(i=0,1,..16)$	flip-flop used in input-output, store state of flip-flop register	core storage
S_I	60	$R_3 \to S_I$		symbol recognizer on input register	flip-flops MK-1 core storage MK-0
D_I	61	$R_3 \to D_I$		bit mask for S_I	flip-flops MK-1 core storage MK-0
S_O	62	$R_3 \to S_O$		symbol recognizer on output register	flip-flops
D_O	63	$R_3 \to D_O$		bit mask for S_O	flip-flops
K_I	70	$R_j \to K_I$ $(j=2,3)$	$K_I \to R_j$ $(j=2,3)$	character counter for input register	flip-flops MK-1 core storage MK-0
K_O	71	$R_O \to K_1$ $(j=2,3)$	$K_O \to R_j$ $(j=2,3)$	character counter for output register	flip-flops
RS_k (Not addressable) $(k=1,2,4,8,16)$		$R^i \to RS_k^{i+k}$ $(i=0,1,..23)$	$RS_k^{i-k} \to R^{i-k}$ $(i=0,1,..23)$	right shift	core storage
LS_k (Not addressable) $(k=1,2,4,8,16)$		$R^i \to LS_k^{i-k}$ $(i=0,1,..23)$	$LS_k^{i-k} \to R^{i-k}$ $(i=0,1,..23)$	left shift	core storage
W_k (Not addressable) $(k=0,1,2,3)$		$R_3 \to W_k$ $(k=0,1,2,3)$	$W_k \to R_k$ $(k=0,1,2,3)$	word assembly	core storage
G_k (Not addressable) $(k=0,1,2,3)$		$R_k \to G_k$ $(k=0,1,2,3)$	$G_k \to R_3$ $(k=0,1,2,3)$	group assembly	core storage
H_k (Not addressable) $(k=0,1,2,3)$		$R_j \to H_{kj}$ $(j=0,1,2,3$ but $j \neq k)$	$H_{kj} \to R_j$ $(j=0,1,2,3$ but $j \neq k)$	stores word without storing group k	core storage
J_0 (Not addressable)		$R_j \to J_{0j}$ $(j=0,1,2,3)$	$J_0 \to R_{J_0}$	wired to set F_{J_0} if all 1's in J_0	core storage
J_1 (Not addressable)		$R_j \to \bar{J}_{1j}$ $(j=0,1,2,3)$	$\bar{J}_1 \to F_{J_1}$	wired to set F_{J_1} if all 1's in J_1	core storage

sign, and changing between 4-bit to 6-bit per character binary representation. The central control unit contains two Telemeter Magnetics, Inc., *1092-BU-7* magnetic core buffers for storage and uses transistorized logical and control circuits. Four magnetic tape handlers are provided to permit flexibility so that conversions may proceed while changing reels. Sufficient flexibility was provided so that new conversions could be handled with a minimum of modification.

The sequence-controlled calculator developed into the automatic data processing machine, and in the same manner it is expected that the sequence-controlled data converter will evolve into a stored program, selective sequence-controlled, special-purpose data-processing machine. This will perform complex code changes such as those required in binary to decimal conversions. Frequently, it will be used to reduce the amount of information fed to other machines by using curve fitting techniques and, in the process, may perform sophisticated forms of error detection and correction. It will handle highly variable field structure within the information and will be capable of performing simple sorting operations. This approach will allow the operator to use data-processing machines to their full efficiency without undue restriction in the form of legislation. This concept is compatible with the present development of the computing art in which different machines perform steps in the flow of data processing. The simple intermediate steps will be performed by a data conversion computer.

Proposed Design of Special-Purpose Computer for Data Conversion

This design adopts the philosophy that a central control unit of a data converter which may be programmed like a stored program general-purpose computer, is more useful and flexible than a sequence-controlled plug-board unit. As the translation problem becomes more complex, it is only necessary to change the program. This is different from the philosophy of adding special-purpose computing circuits as new translations occur.

The generalized central control unit (GCCU) has the facility of controlling adapters and internal operations by stored program commands and a plugboard for changing the interpretation of input-output controls to adapters. It contains a high-speed random access core memory called internal memory (IM) for internal storage, (512 24-bit words), two sequential buffer core memories (*TMI 1092-BU-7*) *BU1* and *BU2*, a specially wired core memory called core storage registers (CS) used for most internal registers required for performing commands, a set of flip-flops for controlling input-output functions, a plug-board to define the relation between these flip-flops and the adapters, internal control circuits, and miscellaneous registers. Basic functions are:

1. To select and control input-output adapters and their mode of operation, such as read or write, forward or backward, rewind, etc.

2. To transfer information between adapters, buffers, IM registers, and the CS registers.

3. To interpret information from the adapters or internally generated so as to change the mode of operation by changing the state of the adapter or by internal transfers. For example, the tape unit may be stopped when an end of block is reached, a certain symbol recognized, or a given count reached.

4. To do stored programmed computing on data entering the system so that a translation may be made from an input language, given its structure, to an output language, given its structure. Typical operations are changing word length and structure, code transformation by table look-up or algorithms, changing block length, editing, collating, sorting, and searching.

INFORMATION FLOW IN THE GENERALIZED CENTRAL CONTROL UNIT

The structure of the GCCU excluding the input-output controls is displayed in

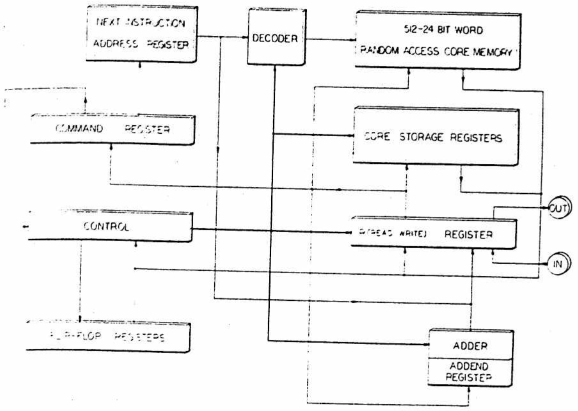


Fig. 4. Generalized central control unit block diagram

Fig. 4. All information flows through the 24-bit R (read-write) register. It serves as the static register for reading, or writing, from, or to, the IM or the CS registers.

The address of the next instruction is stored in the next instruction address register (NIA) which is decoded to select the instruction from the internal memory. The word in the selected address is read into the R register and then transferred to the command (C) register.

Most commands are executed by transfers between information in the CS and the IM. The read-write time for random access to a word in the IM is 6 microseconds. The read and write times for selected CSR are 3 microseconds each. Any commands involving arithmetic operations utilize the 12-bit parallel adder. Individual flip-flops in the flip-flop registers which control input-output operations may be set and reset either by commands or from other information sources. Information from input-output enters and leaves the IM and CS register through the last six bits of the R register.

A list of the commands, their bit structure, the functions performed, and execution times may be found in Table I. The 24-bit word may be considered as four 6-bit groups. The first group usually contains the two octal digits of the command code. The second group controls index registers and miscellaneous operations. The last two groups usually contain the address involved in the operation with the last two bits of the address referring to one of the four groups of a word when pertinent.

STRUCTURE OF THE CORE STORAGE REGISTERS

One of the unique features of the GCCU is the way the CS registers are utilized in performing instructions. A complete list of functions of the registers and their wiring may be found in Table II. Their full use in the commands will be discussed later. Some typical registers will be discussed.

Fig. 5 shows schematic wiring of the LS_1 (left shift 1), A (accumulator) and RS_1 (right shift 1) CS registers. Information is transferred between the R register and the selected CS register by writing into the selected register from R and reading the selected register into R. The wiring of the cores for each register determines into which bit positions the information is to go. The wires with arrows pointing upward refer to writing in the cores. The wires with arrows pointing downward refer to reading from the cores.

When writing into the A register, bit R_j^i (i is a bit in group j) is written into a core with the same bit position when A is selected. When reading, this core is wired back to bit R_j^i. The A register merely stores information without change. When the RS_1 register is selected, information written from bit R_j^i is read into bit R_j^{i+1}. Thus a shift of one bit to the right is accomplished.

The series of transfers required to shift the A register one bit to the right would consist of:

$A \rightarrow R$

$R \rightarrow RS_1$

$RS_1 \rightarrow R$

$R \rightarrow A$

The shift commands are mechanized by transfers of this type to registers RS_1, RS_2, RS_4, RS_8, and RS_{16}. A shift of binary 10110 (22) to the right utilizes registers RS_2, RS_4, and RS_{16} in executing the command.

Fig. 6 shows the wiring of the word assembly (W_i) and group assembly (G_i) registers. Since information is to enter the IM from the buffers in 6-bit characters, it is necessary that these characters be assembled into 24-bit words. Words consist of 24 bits or four groups of six bits. The figure shows the wiring of one bit from each group. The W_i registers each contain six consecutive cores representing the six bits of the selected group. The groups entering will be

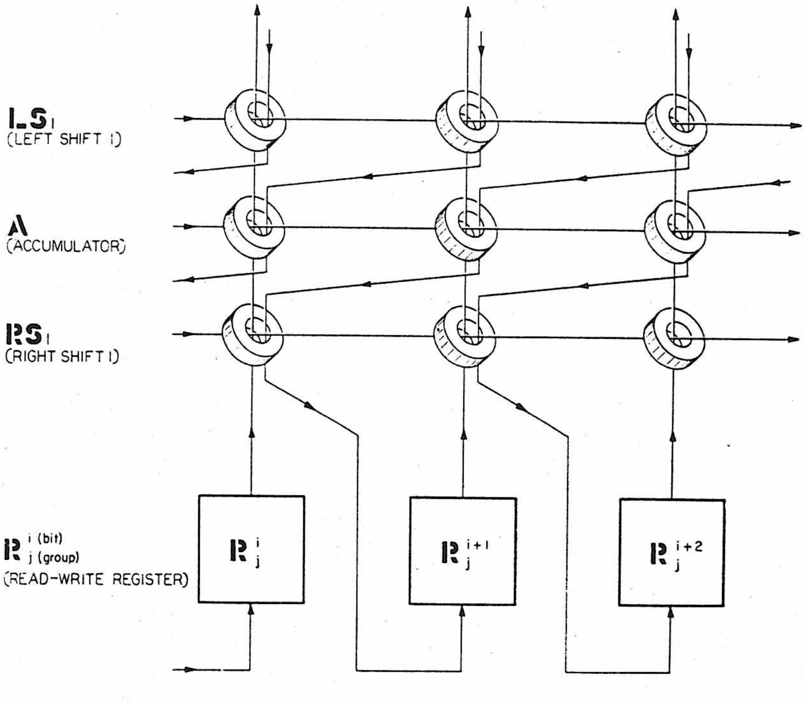


Fig. 5. Input-output selection and information flow (GCCU MK-O)

TABLE III. EXAMPLE OF MODE CONDITIONING COMMAND

7	4	μ_0	μ_1	μ_2	μ_3
1 1 1	1 0 0	1 1 0 1 1 0	1 0 0	1 0 0	1 1 0 1 1 0

μ_0: Selects Fixed Mode of Operation for Selected Unit

μ_o^o = Tape Motion Forward or Backward [Fwd.]
μ_o^1 = Read or Write Tape [Read]
μ_o^2 = No Function
μ_o^3 = Write Density 1 or Density 2 [Density 1]
$\left.\begin{matrix}\mu_o^4 = \\ \mu_o^5 =\end{matrix}\right\}$ Control Interblock Spacing

μ_1: Selects Variable Mode of Operation for Selected Unit

μ_1^o = Start or Stop [Start]
μ_1^1 = Fast Rewind
μ_1^2 = No Function

μ_2: Selects μ_1 Modes to Change if any Flip-Flops Selected in μ_3 Set

μ_2^o = 1 (μ_1^o changes state if any Flip-flops in μ_3 set)
μ_2^1 = 0 (μ_1^1 does not change state)
μ_2^2 = 0

μ_3: Indicates a Change in Mode of Information Concerning Adapter

μ_3^o = End of Block
μ_3^1 = Selected Count of Characters Exceeded
μ_3^2 = Symbol Recognized
μ_3^3 = End of Tape
μ_3^4 = End of File
μ_3^5 = No Function

called γ_0, γ_1, γ_2, and γ_3, and the desired word is the string $\gamma_1\ \gamma_2\ \gamma_3\ \gamma_4$. Each group enters group 3 of R (R_3) in sequence with the bits of each group transferred in parallel. The sequence of transfers necessary to assemble a word consist of $\gamma_0 \rightarrow R_3$, $R_3 \rightarrow W_0$; $\gamma_1 \rightarrow R_3$, $R_5 \rightarrow W_1$; $\gamma_2 \rightarrow R_3$, $R_3 \rightarrow W_2$; $\gamma_3 \rightarrow R_3$. $R_3 \rightarrow W_0$; and finally the simultaneous transfer of $W_i \rightarrow R_i$ (i = 0,1,2,3,). Thus the assembled word ends up in the R register and may be written into memory.

When a word is to be read from memory to a buffer a character at a time, the CS registers G_i (i = 0,1,2,3) are used. The simultaneous transfer $R_i \rightarrow G_i$ (i = 0,1,2,3) occurs first, then the sequential group transfers $G_i \rightarrow R_3$, $R_3 \rightarrow$ buffer (i increasing from 0 to 3).

Command codes which have their first octal digits a 0,1, or 2 control transfers between certain addressable numbered CS registers R_i (see Table II) (i — 1,2,. . .73) and IM and among themselves. Typical of these addressable registers are those which serve the function of index registers, accumulator, quotient, extractors, and complementor.

Other CS registers, which are not explicitly addressable in these commands, such as the registers used for shifting and assembly are used in executing other commands. Commands 40 and 41 are typical in that they use special core registers to select any group from the contents of an index-modified address and place it in a selected group of A or Q (quotient register) and vice versa.

STRUCTURE OF THE INPUT-OUTPUT CONTROLS

The structure of these controls allow the GCCU to achieve the versatility required in performing data conversions. In the process of designing the input-output controls, two systems of information flow emerged. *MKO*, Fig. 7, the simple system is less versatile, and less expensive since it uses one input-output register. *MK-1*, Fig. 8, uses two registers, one for input and one for output allowing a very versatile system, but more expensive.

Command code 70 is used in selecting information flow paths. It is divided into five parts controlling five information paths denoted by ϕ_i (i = 0,1,2,3,4). The command is executed by setting flip-flops which select the chosen paths. These paths remain until they are changed by another 70 command with different ϕ_i's. The amount of information flow, when it starts and stops, is controlled by commands 74 and 75.

MK-0, Fig. 7, only uses four ϕ's. ϕ_0 controls information flow from the adapters, buffers, or R register to the $I/0$ register. The superscript on ϕ_0 denotes which particular device is selected for transmission. ϕ_1 controls information from $I/0$ to a selected adapter ϕ_1^j (j = 4 to 63). ϕ_2 controls information from $I/0$ to a selected buffer ϕ_2^k, (k = 1,2). ϕ_3 controls information from $I/0$ to the R register.

Two or more ϕ's in command 70 set up an information flow path. If a numerical zero is in a given ϕ field, that ϕ is not selected. Information normally flows from a selected adapter i to a buffer. Thus if one block is to be transferred into buffer 1, ϕ_0^i and ϕ_2^1 would be selected. The information is processed by taking information from buffer 1 to IM by selecting ϕ_0^1 and ϕ_3^3. If it is desired to recirculate the information in buffer 1, then ϕ^1_2 is also selected. The processed information may then be sent to buffer 2 by selecting ϕ_0^3 and ϕ_2^2. It probably would then be sent out to adapter j by selecting ϕ_0^2 and ϕ_1^j.

MK-0 thus has the limitation that only one process such as inputting, outputting, or computing is going on at any one time. Except for the loss in time in getting things done this isn't too serious. However, it does limit the ability to input from a real time source which is generating data at a constant rate and cannot be stopped without losing data.

MK-1, Fig. 8, uses five flow path selectors ϕ_i. ϕ_0, ϕ_1, and ϕ_2 control information flow similar to *MK-0* except that the input and output registers are separated. ϕ_3 controls information from the input register to the R register or a selected buffer. ϕ_4 controls information from a selected buffer or the R register to the output register.

Considerable flexibility is possible now. Information may be flowing to buffer 1 from adapter i, and information may be going to the R register from buffer 2 by

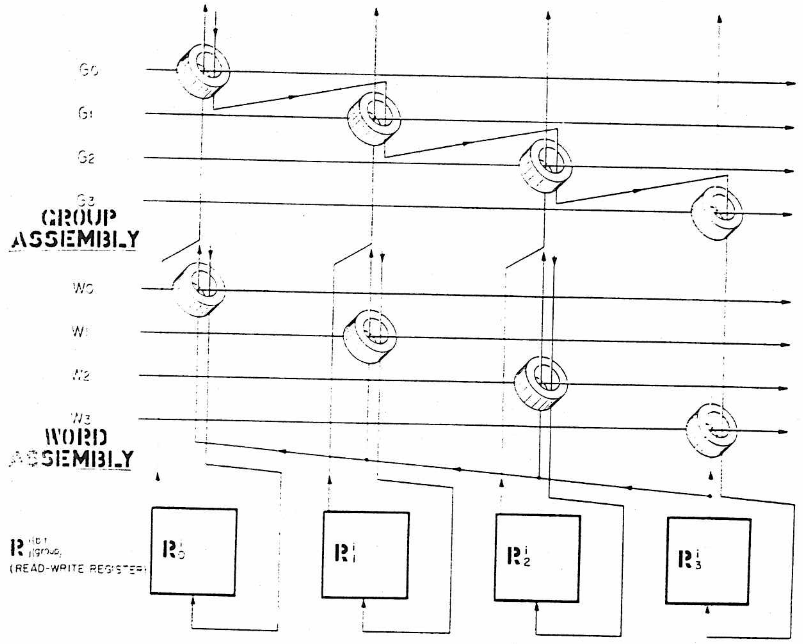


Fig. 6. Input-output selection and information flow (GCCU MK-1)

selecting ϕ_0^i, ϕ_3^1, ϕ_4^2, and ϕ_1^3. Inputting and outputting, while recirculating a buffer, may be done by selecting ϕ_0^i, ϕ_3^2, ϕ_4^1, ϕ_2^1, and ϕ_1^i. It is possible to output from buffer 1 and transfer information back and forth from buffer 2 to the R register by selecting ϕ_4^1, ϕ_1^i, ϕ_0^3, ϕ_3^2, and then selecting ϕ_4^1, ϕ_1^i, ϕ_0^2, ϕ_3^3.

All the combinations of paths will not be enumerated here, since they are too numerous. It is interesting to note that the roles of adapters, buffers, and the R register are similar in being transmitters or receivers through the input and output registers. Thus, transfers between internal units, buffers and R register, are controlled like transfers between internal and external units. The limitations imposed by using the *MK-0* scheme no longer exist. Real-time continuous inputs may be used provided that there is time to process and output the data. Computing may be going on simultaneously with inputting or outputting. There is a gain not only in being able to do more functions, but a real gain in getting the conversion job done much faster.

The valves of input-output information are controlled by the mode conditioning commands 74 and 75. These commands select a mode of operation for an input-output unit, and determine how the mode will change dependent upon other selected conditions. The input-output plug-board determines specific relations between selected bits of the command and properties of the input-output units. The second octal digit 4 or 5 of the command code determines which flow path is being controlled. 74 controls ϕ_0 selected paths and 75 controls ϕ_1 selected paths. Only these two paths need to be controlled. It is only necessary to establish that either the generator or the receiver controls the basic timing.

The command consists of four parts μ_i ($i = 0,1,2,3$). An example of a command is given in Table III. μ_0 selects the fixed mode of operation for the selected units; for example read or write, forward or backward, and bit density. Modes selected here remain fixed until a new command is given changing them. μ_1 selects variable modes of operation which may be changed conditional on bits selected in μ_2 and μ_3. μ_2 selects which bits of μ_1 will be changed, and μ_3 selects the conditions for change. For example a bit in μ_1 may control start or stop. If the corresponding bit in μ_2 is selected, then the unit will change from start to stop when conditions selected in μ_3 occur. The μ_3 condition might be that an interblock gap occurs, or a certain number of characters has entered, or a symbol has been recognized, etc.

The plug-board wiring determines which particular characteristics of the selected unit are to be connected with bit positions of the command, and which flip-flops of the flip-flop registers are to be used for the bit positions. For example, when adapter i is used, its forward or backward control is connected to μ_0^0, and the flip-flop which controls might be F_{012}. Adapter j might have its forward or backward control connected to μ_0^0 also, but the flip-flop which controls might be F_{013}. Thus adapter i could be inputting, while adapter j is outputting.

The number of characters entering the I register will be counted in the K_I counter, and also those leaving the O register will be counted in the K_O counter. Fixed counts may be selected by presetting the counters since the counters are addressable through the register transfer commands (first octal digit 0,1, or 2).

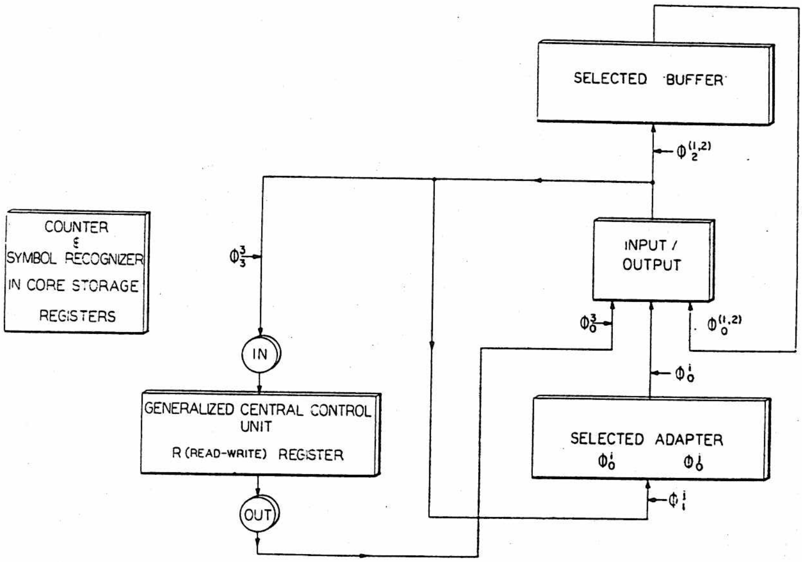


Fig. 7. Core storage registers LS_1, A, and RS_1, wiring

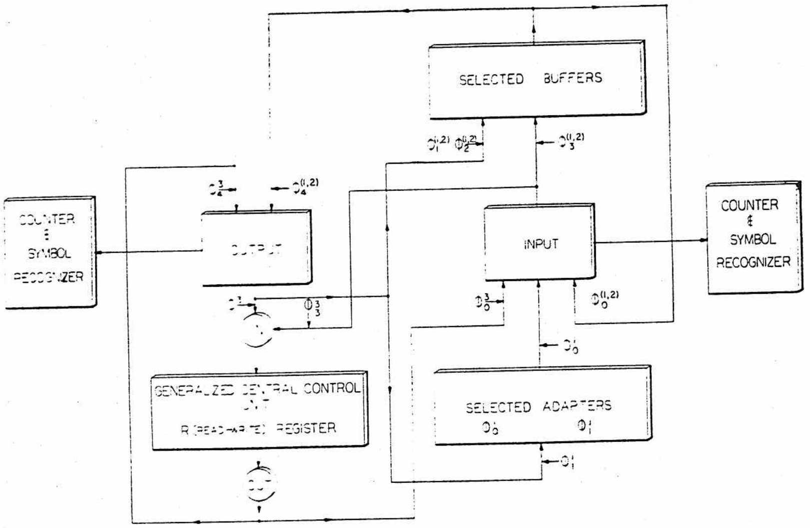


Fig. 8. Core storage registers G_i and W_i wiring

Selected flip-flops will be set when the counter goes through zero.

Arbitrary symbols may be recognized when passing through the I or O register. This is accomplished with the addressable registers S_I, D_I and S_O, D_O. S_I and S_O contain the symbol and D_I and D_O contain those bits of the symbol which are to be compared. This was included since it is not always desirable to compare all six bits of a character for recognition.

The Telemeter Magnetics, Inc. *1092-BU-7* sequential buffers have the property that a pulse is emitted when the buffer is full or empty. These pulses may be used to set flip-flops by using the plug-board position associated with the two internal buffers. If block sizes greater than *1092* are required, standard *2184-BU-7* buffers may be used instead of the *1092-BU-7*. It should be pointed out that additional memory may be added to the GCCU by considering additional buffers as input-output equipment. Similarly storage devices of large capacity such as magnetic drums may be added as input-output devices if needed for some particular applications.

Other Commands

The other commands not previously discussed relate to index registers, jumps, flip-flop controls, arithmetic operations, and extracts. These commands are listed in Table I; however, they will be discussed briefly to complete the picture of the GCCU.

There are seven index registers B_i ($i = 1,2,\ldots,7$) contained in the CS registers. Commands which may be index-modified contain a B_i in the last three bits of group 1. Transfers to and from B_i's are controlled by the transfer commands. The contents of an index register may be modified by adding N to it or by adding the contents of an index-modified address. A jump instruction is also provided so that a jump to an index-modified address is made if the contents of the selected index register is zero.

There are conditional jumps made on the contents of the A or Q registers being zero, and an unconditional jump. Index-modified address jumps may also be made on the set condition of any of the 64 addressable flip-flops. Groups of flip-flops selected by bit position in a word or by address may be set or reset. The use of these flip-flop commands allows the programmer to program break-points, since flip-flops can be set by switches. When trouble-shooting programs, the contents of IM or the CS register may be examined by programming the GCCU to read out the information of interest on an automatic typewriter.

The arithmetic operations provided are add, subtract, multiply, and divide. Addition and subtraction may be done by adding and subtracting the contents of an index-modified address to either the A or Q register, with the sum remaining in the register. Multiplication and division are performed with the multiplicand (or dividend) in A, and the multiplier (or divisor) in an index-modified address. The most significant part of the product appears in A, and the least significant part in Q. The quotient appears in Q and the remainder in A. These operations may be performed on the whole word, the last 12 bits or the last 6 bits (group 3). The representation of negative numbers is by 1's complement.

There is an extract command which forms the logical product of the selected extract register with the contents of an index-modified address, and which then forms the logical sum of this product with A or Q. The extract registers E_i ($i = 0,1,2,3$) are addressable CS registers.

Applications of the Generalized Central Control Unit

The GCCU will do as fast a job as a media translator or a data converter with a conventional CCU will do. It will do much more since programs can be written which will take care of any translation problem which can be programmed. If new adapters need to be added to take care of new physical format problems, they are added by means of the plug-board. But the internal machine does not change, even if new linguistic problems arise.

The time required to do a conversion of course depends on the complexity of the operations, and the capabilities of the programmer. An IBM *705* to Remington Rand Univac *1103A* conversion program was written. The conversion rules were that variable IBM *705* words were to be made into fixed length *1103A* words by chopping long words and adding zero to short words. The binary coded decimal word was to be converted to a binary word. The variable IBM record was to be made into a fixed Remington Rand Univac block.

All the housekeeping, counting, and rearranging took 75 words in IM, including instructions and storage of constants. The binary-coded decimal to binary conversion subroutine utilized a table of the double-precision binary equivalents of the $b\ c\ d$ digits for the appropriate powers of ten. This takes 9 ($b\ c\ d$ character/power of 10) times [7 (powers of 10) times 2 (double precision binary equivalents) + 4 (powers of 10) times 1 (single precision binary equivalents)] or 162 words in IM. The double-precision binary equivalent is found by summing the equivalents of each $b\ c\ d$ digit found in the table. In addition to the table, 56 words were required to program the subroutine. This method of table look-up is the most efficient timewise technique to use for the GCCU, since the machine was designed to handle manipulation of data much more rapidly than its arithmetic operations.

Not much has been said of other adapters beside those used for magnetic tape units. Some important adapters which have been considered are these which allow for punched card and punched paper tape inputs, and for punched card, punched paper tape, and high-speed printer output. In the punched card and printer cases, special magnetic core buffers would be used in the adapters for the special conversions required. In the case of punched card input, a Telemeter Magnetics, Inc. *80-CB-7* magnetic core buffer would be used. This takes information presented in 80 columns, 12 rows and presents it with some decoding into 6 bit serial characters. For punched card output, a Telemeter Magnetics, Inc. *80-CB-12* magnetic core buffer would be used. This takes 6-bit serial character information and presents 12-row 80-bit parallel output. A typical printer might use a *120-BA-56* which takes 6-bit serial character information and forms a print wheel core image for printing.

Besides doing conversions with difficult variable linguistic format transformations, this equipment is well suited to do complex editing searches, collating and sorting. Simple editing, like selecting given characters out of given blocks, is done at speeds comparable with pluggable data converters with unlimited selection capabilities.

Complex editing used when selecting only significant data by criteria such as selecting significant points for curve fitting take more time, but can be done efficiently if not too many arithmetic operations are required. Searches, collating and sorting are done rapidly since the equipment was designed to handle table look-ups, comparisons, and manipulation of data rapidly.

REMINGTON RAND UNIVAC *1103A*—IBM *704* MEDIA TRANSLATOR FUNCTIONS

The functions of the Remington Rand Univac *1103A*—IBM *704* media, translator follow:

A. Conversion
  1. Remington Rand Univac 1103A magnetic tape and IBM 704 magnetic tape.

B. Functions
  1. Checks Remington Rand Univac *1103A* magnetic tape for lateral parity and block length.
  2. Checks IBM *704* magnetic tape for lateral and longitudinal parity.
  3. Fixed block length conversions from blocks of 60 characters to 1,000.
  4. Input may be continuous up to 5 kc (without inter-block gaps) and output in block format.
  5. Bad spot logic and control for Remington Rand tape.
  6. Block counters to position tape or translate a preset number of blocks.
  7. IBM end of file mark stops conversion optionally.
  8. Speed limited only by tape handlers.
  9. Newly written tapes may be checked on rewind.
  10. End of tape mark can be written manually.

AIR FORCE MISSILE TEST CENTER DATA CONVERTER FUNCTIONS

A. Conversions
  1. FLAC magnetic tape⇄FLAC paper tape.
  2. IBM (*704, 650*) magnetic tape⇄FLAC paper tape.
  3. Remington Rand Univac *1103A* paper tape⇄FLAC paper tape.
  4. FLAC paper tape⇄FLAC paper tape.
  5. IBM (*704, 650*) magnetic tape⇄FLAC magnetic tape.
  6. Remington Rand Univac *1103A* paper tape⇄FLAC magnetic tape.
  7. FLAC magnetic tape⇄FLAC magnetic tape.

B. Pluggable or Switch Operations
  1. Odd or even lateral parity check and generation.
  2. Ten character codes may be recognized and used for control.
  3. Parts of blocks may be converted by counter selection:
    (a) Select any four words in block.
    (b) Select all words between counter selected boundaries.
  4. Block counter used to convert every *n*th selected block.
  5. Fixed word length from variable word length by clipping or adding zeros.
  6. Automatic complementing and sign code conversion from *1103A* paper tape.
  7. FLAC to IBM *650* causes a change of position of sign in word.
  8. A word of six 6-bit characters may be converted to a word of ten 4-bit characters and vice versa.

1958 western joint

The Universal Data Transcriber. A New Approach to Data Conversion Equipment

MARVIN S. MAXWELL
NONMEMBER AIEE

THE Universal Data Transcriber is a device to convert digital data from one form to another. The input to the system can be cards, paper tape, magnetic tape, or the output of almost any source of digital data that does not exceed 500,000 bits per second. The output of the converter can be cards, magnetic tape, paper tape, low- or high-speed printers, point plotters, or almost any other digital output device. The conversion process allows for very flexible code and format conversion as the data is transcribed from one medium to another.

The Universal Data Transcriber consists of appropriate input-output equipment and a microprogrammed, semi-variable-structure, stored-programmed, single-address binary computer. The computer has a high-speed random access memory of 8,192 characters of 8 data bits plus one check bit each, and will average 70,000 instructions per second. The operation of the system depends upon the microprogram of the computer to generate special orders which will transfer data from the particular external input device currently in use to the computer memory, and from the memory to the external output device currently in use. The use of microprogramming, which is accomplished by use of a plugboard, allows an efficient transfer of data between the computer memory and the external devices with a minimum of special equipment. Conversion of the data within the memory from one form to another is accomplished by the use of an appropriate stored program. This gives a very flexible system, since all that is required to change the system from one job to another is to change the connections to the external equipment, insert a different plugboard, and load a new program into the computer memory. This system was conceived, designed, and is under construction by the Computer Research and Development Branch of the Computation and Exterior Ballistics Laboratory of the United States Naval Proving Ground, Dahlgren, Virginia.

Description of the Computer

The basic structure of the computer is shown in the diagram of Fig. 1. It con-

---

MARVIN S. MAXWELL is with the United States Naval Proving Ground, Dahlgren, Va.

A project of this nature is the work of a number of people. The author wishes to thank P. H. Burkhart for his assistance in writing this report, W. P. Warner for his assistance in preparing preliminary programs, and J. Pinker, Jr., G. V. Olds, A. O. Danello, J. C. Carlock, E. J. Culhan, and J. R. Gros for their work in designing the system.

sists of an input register, output register, two computing registers (R-1 and R-2), six B-registers (address modifiers), instruction register, instruction counter, indicator latches (single-bit registers), and other special registers. External devices communicate with the computer via the input and output registers under control of the computer. The input register can select at high speed from either of two different external devices. The output register is normally connected to only one unit. Indicator latches are used both to control the external devices and to signal the condition of the external devices to the computer. Special electronic signal-generating equipment tailored to each type of external device is used to facilitate communication with the input register, output register, indicator latches, and the external device.

The computer has a high-speed random access core memory with a capacity of 8,192 characters, each consisting of eight bits plus a check bit. A character is treated as a pure binary number by the arithmetic section of the computer and a single instruction generally affects only one character. Since there are no multiply or divide orders, the operating binary point may be considered to be in any convenient location. The carry (borrow) bit may be propagated from character to character in addition (subtraction) with use of double precision orders. A single reference to the memory brings out four characters designated as $M0$, $M1$, $M2$, and $M3$ into the memory register. Addresses evenly divisible by four always correspond to the character read out as $M0$. Instruction words consist of the four characters $M0$, $M1$, $M2$, and $M3$. Instruction words are logically divided into 4 fields as shown, namely: operation code, B-register specification, address specification of reference to memory, and the limit value of Bx.

The time required to execute orders which require only one reference to memory such as program transfers is approximately 11 microseconds. Orders which require two references to memory, such as read out or store orders, require approximately 21 microseconds. A complete description of the order codes is available in Appendix I.

Each of the six B-registers has eight binary columns plus a check column. Address modification is performed by overlaying the contents of the eight least significant columns of the 13 binary-column address field with the contents of the specified B-register. The overlaying process is a logical addition in which the contents of a binary column in the address field is changed only when a zero is overlaid with a one-bit.

For example:

Address field	1	1011	1001	0110
B-register			0101	0011
Result	1	1011	1101	0111

The contents of B-registers may be incremented by one either automatically after use if so desired or on command. If it is desired to increase the contents of a B-register by any value other than one or to decrease its contents by any value, the B-register may be operated on by the use of appropriate order codes.

One of the B-registers, $B6$, is normally connected as a real-time counter. It may be preset to any value up to its limit of eight bits and a one is added to the contents of the register every 0.504 millisecond. The counting ceases when the contents of the register reaches 377 octal. With use of this feature and instructions for read out and limit sensing on the B-registers it is possible to program measurement of real-time for controlling external devices.

	Bits	
Operation code	8 7 6 5 4 3 2 1	$M0$
B-register specifications	8 7 6 5 4 3 2 1	$M1$
Address specification of reference to memory	8 7 6 5 4 3 2 1	$M2$
Limit value of Bx	8 7 6 5 4 3 2 1	$M3$

CHECKING PROCEDURES

The computer has automatic circuitry built into the system to check the accuracy of its operation. This check adds a parity bit to the 8 bits in each character so that the modulo two sum of the binary one's of these nine bits is always odd. This check bit is generated after data enters the input register, is corrected as the characters are modified by various orders, and is stored in the memory along with the character. An automatic check is made for the presence of the proper parity count as the data are transferred from the memory into the working registers or the instruction register. The values in the B-registers are checked automatically as they are used and there are checks on the execution of the overlay and shifting operations in the computing registers.

Whenever possible, checks will be made on the accuracy of data transmission between the computer and the external devices. For example, in card reading, data will be loaded into two independent shift registers from two reading stations, and after the card images are assembled in memory they will be checked against each other. In punching data into cards, the card will be read back into the computer after being punched, and this card image will be checked against the card image sent out to the punch. When magnetic tapes are written, the data will be read back into the computer and a check will be made on the correctness of the data.

Special Features of the UDT

The most outstanding difference between the computer of the Universal Data Transcriber (UDT) and any other single-address binary computer is the availability of the plugboard and the plugboard instructions. The plugboard is divided into three regions. The first region consists of information coming from equipment in the computer to the plugboard. This includes all of the registers, such as register 1, register 2, input register, output register, instruction register, instruction counter, $B7$, and the indicator latches, plugboard instruction specification, and the internal clock. Also in this region are external inputs from the various input and output devices which have been converted to the proper signal levels. The second region of the plugboard consists of a set of approximately 75 logical packages. These packages are identical to those used in the construction of the rest of the computer. In the third region of the plugboard are exits from the plugboard to the control lines in the computer. These lines control the transfer of data from "register to register," use of the B-registers, controlling memory cycles, setting of indicator latches, shifting various registers, etc. Thus, by using all

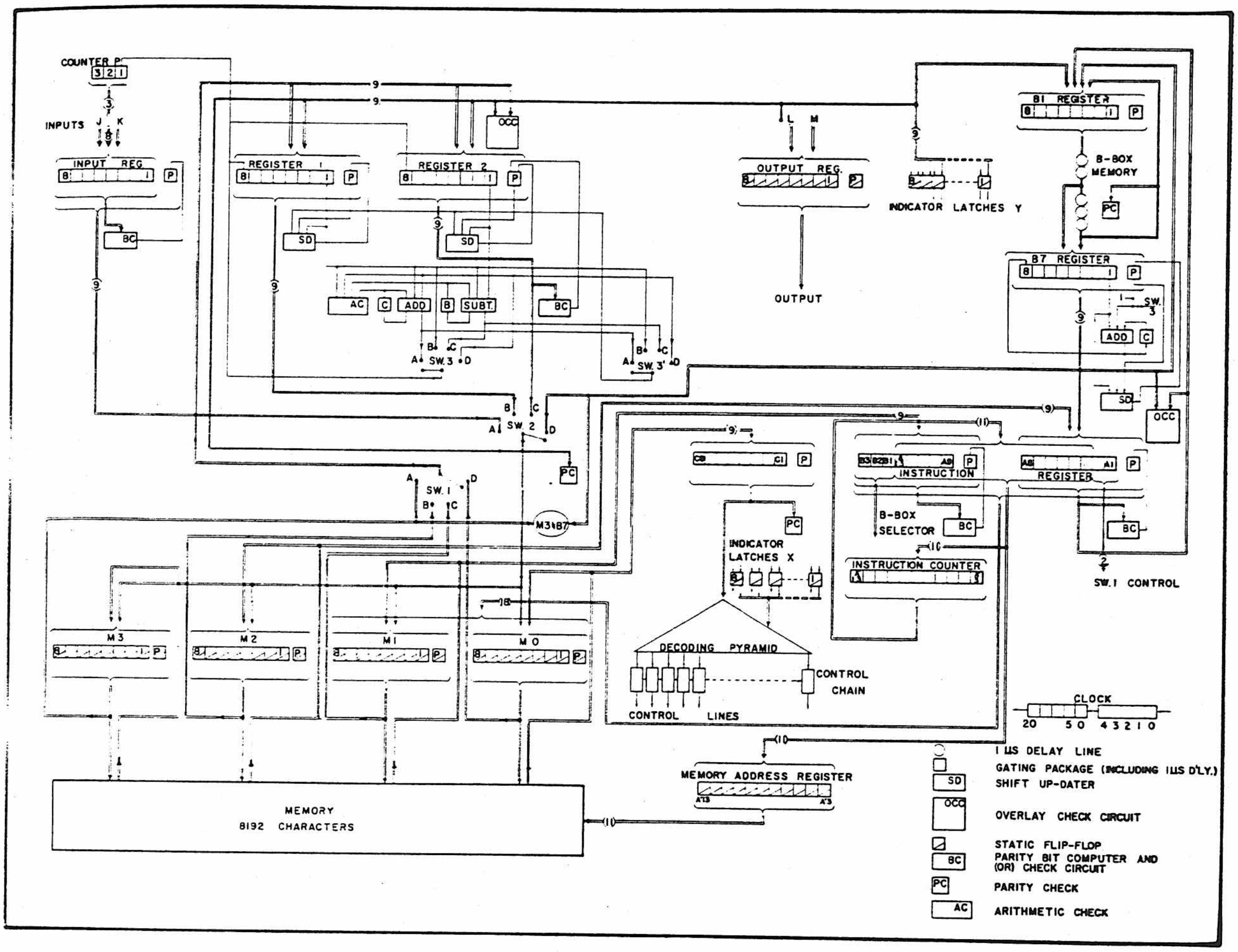


Fig. 1. Block diagram of Universal Data Transcriber

three regions of the plugboard almost any conceivable, or desirable, cycle of actions can be controlled from the plugboard. This feature is primarily for use with external devices to get data to or from them and the memory of the UDT.

The indicator latches in the computer are used primarily for communication between the UDT and external devices. For example, some of the indicator latches could be wired, via the plugboard, to control the stopping, starting, or reading or writing of a tape unit. Other indicator latches could be used to indicate to the UDT that an external device is in certain conditions; for example, that a card reader is moving cards, or ready to scan one row of information, or that it is out of cards, etc. Thus, the program can control external devices, and external devices can be sensed by the program by use of the indicator latches.

Another feature of the UDT is the "program interrupt" ability. If a particular exit on the plugboard is energized the computer will go into a program interrupt cycle. This exit can be energized from an indicator latch, or combinations of indicator latches and various conditions by wiring on the plugboard. When this condition occurs the computer will automatically make a program transfer to instruction location 4 at the end of the current instruction. The address, Y, of the instruction which would have normally been executed next, if the program interrupt condition had not occurred, will be automatically stored in character locations 1 and 2 in a form so that if the character in location 0 is the code for a program transfer, jump, command and the instruction at location 0 were to be executed, the computer would jump to the proper address, Y. When this feature is used the program, starting at location 4, must be suitable to take the appropriate action for the condition which caused the jump. After this is done, the program would normally remake the appropriate registers, and then jump to location 0, which would cause the jump back to the main program at the proper place. By using this feature the computer can react rapidly to external control information without requiring repeated sensing on the condition.

Advantages and Limitations

The major advantage of the UDT is its flexibility. It is not tailored to any specific computer or type of data conversion and is therefore not likely to become obsolete as fast as many specialized converters. The microprogramming and stored program features makes it easy to implement almost any desired conversion with a minimum of engineering effort and special equipment. The major disadvantage of this approach is that it is more expensive than any single specialized converter.

To establish the capabilities of the UDT, several preliminary programs have been prepared. One program for converting 80-column alphanumeric International Business Machines Corporation (IBM) cards to NORC magnetic tape provides for arbitrary code and format conversion, specified by header cards, and converts data to magnetic tape at a rate of 450 cards per minute. Similar programs have been developed for conversion from one magnetic tape system to another. If there is a conversion in both the code representation of the data and in the format, but not in the number base, the system can convert 4-, 5-, 6-, 7-, or 8-bit characters from one form to another at a rate of approximately 3,000 characters per second. Conversion can be made from 48-bit binary words to decimal digit words at a rate of approximately 16 words per second. Conversion can be made from 13-digit decimal words to binary words at rates in excess of 50 words per second.

Engineering Considerations

The Universal Data Transcriber is being designed and constructed at the United States Naval Proving Ground, Dahlgren, Virginia. Subcontractors are providing the memory, logical building blocks, and various specialized input and output circuitry. The system which is currently being assembled is scheduled to become operational in the first quarter of 1959 and will consist of the central computer, a magnetic tape transport, an IBM *514* used as a card punch, an IBM *407* used as a printer, a modified IBM *101* used as a card reader, and a Flexowriter. The computer with memory, level conversion circuitry, plugboard, power supplies, and console will occupy 5 relay racks.

The logical building blocks are all transistorized megacycle SEAC type circuitry built by Computer Control Company. Some of these are being modified to provide 2-phase operation where the extra speed is required. The memory is an all-transistorized magnetic core memory with a full read-write cycle time of 10 microseconds, and operates in parallel on a 36-bit word or 4 characters of 9 bits each. The 80-brush reading station of the IBM *101*, used as a 450-card per minute reader, will load the data from a row in the card in parallel into a magnetic shift register which will be shifted into the computer on four wires in 600 microseconds. A similar circuit will be used on the second reading station so as to provide a check on the reading. Data are punched into IBM cards at 100 cards per minute by serially shifting, one bit at a time, at a 100,000 cycle shift rate, the 80 bits in the row to be punched. This shift register will pick up relays which will control the punch magnets in an IBM *514*. The reading station which follows the punching station will be equipped with magnetic shift register for reading back the data from the punched card for a check. The same shift register and relays which are used in punching is 120-bits long so that it can be used to control the printing on an IBM *407*. A Flexowriter is permanently attached to the system to provide communication between the computer and the operator and is used as an input for the program tapes, and as an input or output of 5-, 6-, 7-, or 8-channel paper tape.

Conclusions

The Universal Data Transcriber is a very flexible data conversion system. This system not only allows conversion of data from almost any digital source into an appropriate form for a general-purpose computer but also reduces the computing time required in the general-purpose computer, since most of the format conversion and editing will be performed in the Universal Data Transcriber.

Appendix I. Order Codes

An instruction word consists of 32 bits. 13 bits are used to designate the address of the operand; this is called the address field and is designated by A. Three bits specify which of the six B-registers, $B1$, $B2$, $B3$, $B4$, $B5$, or $B6$, (also called B box, index register, or address modifier) is to be used. In all cases the contents of the selected B-register (8 bits) is overlaid, forming the logical sum, with the low-order 8 bits of the A field before the address is used. Eight bits are used to designate the limit value of the contents of the selected B-register to control some conditional transfers. Eight bits, divided into 3 digits, are the command field. The high-order digit specifies the code group 0, 1, 2, or 3. An instruction word is assembled from 4 characters as indicated.

The prime symbol "′" is used to designate "the contents of." For example, a reference to A refers to the 13-bit address field itself, but a reference to A' refers to the character stored at address A. In all cases, unless specifically stated to the contrary, when data is transferred from one register to another the data remain in the source register and the receiving register is cleared to all zeros before, or as, the data are received.

In writing the orders in the following groups the subject matter is consistent with the concepts expressed in the following definitions.

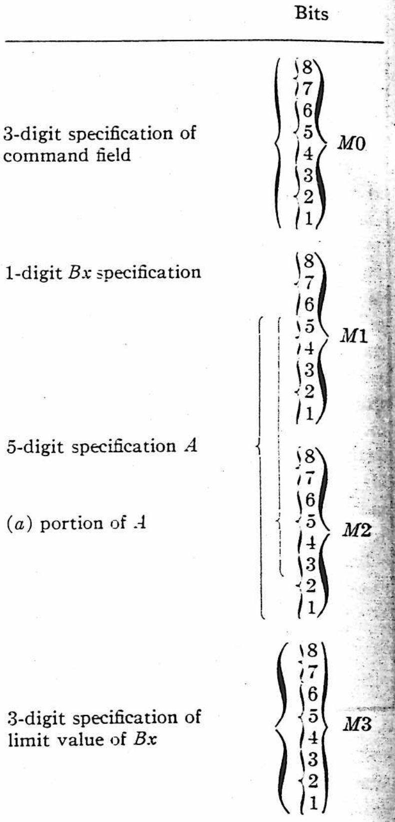
(a) = The 11 high-order bits in a 13-bit address in the instruction register in which the 2 low-order bits are always zero. (a) identifies the location of a specific instruction stored in memory.

A = The 13-bit address in the instruction register which identifies a specific location in the memory.

A' = The 8-bit character made available by access to the memory at address A.

Bx' = The 8-bit character made available by access to that B-box which is designated as Bx.

The address A may be modified by overlaying it with Bx'. A' will then identify the 8-bit character made available by access to the memory at the modified address A.



Group 0

The execution time of orders in this group is 20 microseconds. All A addresses may be overlaid with the contents of any one of six B registers (in the recirculating ring) by specifying Bx in the B-field of the instruction. The value of x may be: 1, 2, 3, 4, 5, or 6 to overlay A with the contents of $B1$, $B2$, $B3$, $B4$, $B5$, or $B6$ respectively.

If $B0$ is specified, there will be no overlay. If $B7$ is specified, the contents of Register $B7$ will not be overlaid on A. Subtract orders do not leave a record of a possible negative sign in the result. Propagate add and subtract orders are used in multiple precision arithmetic.

041 A' to $R1$: clear register 1, transfer A' into register 1.

051 A' to $R1$: add 1 to Bx after modifying A.

021 A' to $R2$: clear register 2, transfer A' into register 2.

031 A' to $R2$; add 1 to Bx after modifying A.

025 Add A' to $R1$: clear register 2, transfer A' into register 2, clear carry storage, add A' to contents of register 1, put sum in register 1, store overflow bit, if any, in carry storage, reset register 2 to all zeros.

035 Add A' to $R1$; add 1 to Bx' after modifying A.

024 Propagate add A' to $R1'$: clear register 2, transfer A' into register 2, add A' to contents of register 1 without previously resetting carry storage, put sum in register 1, store overflow bit, if any, in carry storage, reset register 2 to all zeros.

034 Propagate add A' to $R1'$; add 1 to Bx' after modifying A.

Note: With reference to orders 023, 033, 022 and 032, if the contents of register 1 are greater than or equal to the contents of register 2 (minuend greater than or equal to subtrahend) the result will be a binary difference with final contents of borrow storage equal to zero. If, however, the contents of register 2 are greater than the contents of register 1 (minuend less than subtrahend) there will finally be a one in borrow storage and the true difference will be equal to the contents of register 1 minus 2^8. In this latter case, by subtracting the final contents of register 1 from zero, the absolute value of the difference between the minuend and the subtrahend can be obtained.

For multiple-precision arithmetic the nonreset of the borrow in orders 022 and 032 is related to preceding orders so that the final presence or absence of borrow refers to the entire sequence of operations. The true difference may be equal to the contents of register 1 minus 2^{16} or 2^{24}, etc.

023 Subtract A' from $R1'$: clear register 2, transfer A' into register 2, clear borrow storage, subtract A' from contents of register 1, store the borrow bit, if any, in borrow storage, store the answer in register 1, reset register 2 to all zeros. (See note.)

033 Subtract A' from $R1'$; add 1 to Bx' after modifying A. (See note.)

022 Propagate subtract A' from $R1'$: clear register 2, transfer A' into register 2, subtract A' from contents of register 1 without resetting borrow storage, store the final borrow bit, if any, in borrow storage, store the answer in register 1, reset register 2 to all zeros. (See note).

032 Propagate subtract A' from $R1'$; add 1 to Bx' after modifying A. (See note.)

026 Logical product (extract) A' and $R1'$: clear register 2, transfer A' into register 2, form the logical product, bit by bit, of A' and contents of register 1, store the result in register 1, clear register 2.

Example:

Contents of register 1 = $R1'$ = 10101010
Contents of register 2 = A' = 00001111

Logical product = 00001010

036 Logical product (extract) A' and $R1'$; add 1 to Bx' after modifying A.

027 Add A' to $R1'$ with no internal carry: reset carry storage, clear register 2, transfer A' into register 2, add A' to contents of register 1 with no internal carry propagation during adding, store the result in register 1, clear register 2. The presence of a final result in register 1 is an indication that the original operands were not equal.

Example:

Original $R1'$ = 01011101
$R2' = A'$ = 10111101

Final $R1'$ = 11100000

037 Add A' to $R1'$ with no internal carry; add 1 to Bx' after modifying A.

020 Overlay $R1'$ with A' and put the result in $R2$: reset register 2, overlay $R1'$ with A' and put the result in register 2.

030 Overlay $R1'$ with A' and put the result in $R2$; add 1 to Bx' after modifying A.

040 Store $R1'$ in A: reset the memory at the address A, transfer the contents of register 1 into memory at address A.

050 Store $R1'$ in A; add 1 to Bx' after modifying A.

060 Store $R2'$ in A: reset the memory at the address A transfer the contents of register 2 into memory at address A.

070 Store $R2'$ in A; add 1 to Bx' after modifying A.

011 Add 1 to Bx' and program transfer to (a) if $A \neq 0$ and final $Bx' \neq M3'$: add 1 to the contents of Bx and store the result in Bx. The program will proceed to the next instruction unless $A \neq 0$ and the final value of $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction in address (a).

010 Add $R1'$ to Bx' and program transfer to (a) if $A \neq 0$ and final $Bx' = M3'$; add the contents of register 1 to the contents of Bx and store the result in Bx. The program will proceed to the next instruction unless $A \neq 0$ and the final value of $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a).

044 Shift the 4 low-order bits of $R1'$ into the relay register and program transfer to (a) if $A \neq 0$ and $Bx' \neq M3'$: right shift the 4 low-order bits of Register 1, bit by bit, into the high end of the 120 bit relay shift register at a rate of 5 microseconds per bit. The original four high-order bits of register 1 will remain in register 1 as the four low order bits. The program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a).

Group 1

The low order digit used in codes in this group is designated by the octal digit x. (x may have the value 0, 1, 2, 3, 4, 5, 6 or 7.) In all cases the number of shifts specified by x and the number of bits which may be thereby counted is equal to $x+1$. All shifts are to the right, that is: high order bits are shifted toward lower order positions. For all orders in this group the program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a). The execution time for orders in this group is 10 microseconds if $A = 0$ and 11 microseconds if $A \neq 0$.

11x Right shift register 1, $x+1$ places: the bits shifted out of register 1 are lost.

12x Right shift register 2, $x+1$ places: the bits shifted out of Register 2 are lost.

13x Right shift the contents of register 1 into register 2: right shift the contents of both register 1 and register 2 so that $x+1$ low-order bits of register 1 are moved into the high-order end of register 2. The $x+1$ low-order digits of register 2 are shifted out and lost.

14x Shift the contents of register 2 into register 1: right shift the contents of both register 2 and register 1 such that $x+1$ low-order bits of register 2 are moved into the high-order end of register 1. The $x+1$ low order bits of register 1 are shifted out and lost.

15x Count bits: right shift the contents of register 1, $x+1$ places, inceasing the contents of counter P by the number of binary ones shifted out of register 1. Counter P counts modulo 8.

Group 2

224 Transfer the 8-low-order bits of A to Bx: reset Bx, transfer the 8-low-order bits of A in the instruction register into Bx. The execution time for this order is 14 microseconds.

225 Transfer the contents of register 1 into Bx: reset Bx, transfer $R1'$ into Bx. The execution time for this order is 10 microseconds.

213 Program transfer to (a) if $Bx' \neq M3'$: the program will proceed to the next instruction unless $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a) regardless of whether $A = 0$ or $A \neq 0$. In this program transfer, (a) is not overlaid with the contents of Bx. The execution time for this order is 11 microseconds.

Note: For all of the following orders in Group 2, the program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3$. For this condition there will be a program transfer to the instruction at address (a). The execution time for the following orders in Group 2 is 10 microseconds when $A = 0$ and 11 microseconds when $A \neq 0$.

221 Set Iy: set the condition of all 8 Iy latches in conformity with the information in register 1. There is a latch for each bit in register 1 and this latch is turned on for a "one" condition in register 1 and turned off for a "zero" condition in register 1.

241 Set counter P to zero.

223 Transfer contents of input register to register 1: reset register 1, and then transfer the contents of the input register into register 1.

226 Transfer contents of Bx to register 1: reset register 1, transfer the contents of Bx into register 1.

232 Transfer the contents of counter P into the input register: reset the input register, transfer the contents of counter A by way of the plugboard into the input register. The plugboard allows flexibility in the location, the bits transferred and the sense of the bits transferred into the input register.

230 Input K to input register: reset the input register and transfer 8 bits by way of input K into the input register. Input K is connected to its signal source through the plugboard.

231 Input J to input register: reset the

THE UNIVERSITY OF CHICAGO LIBRARY

input register and transfer 8 bits by way of input J into the input register. Input J is connected to signal source by plugboard.

260 Transfer the contents of register 2 to the output register by way of L: reset the output register, transfer the contents of register 2 to the output register by way of the plugboard and entry terminal L of the output register.

261 Transfer contents of register 2 to the output register by way of M: reset the output register, transfer the contents of register 2 to the output register by way of the plugboard and entry terminal M of the output register.

Group 3

The execution time for orders in this group is 10 microseconds except for order 34x. In the jump orders the eight low-order bits of A may be overlaid with the contents of Bx. When the low-order digit of the code is designated by x, x can have the value of 0, 1, 2, 3, 4, 5, 6, or 7.

310 Jump to (a): unconditionql program transfer to the address (a) specified in the instruction register.

317 Jump to (a) and stop: unconditional program transfer to the address (a) specified in the instruction register and stop before the execution the new instruction.

311 Jump if contents of register $1=0$: conditional program transfer to the address (a) specified in the instruction register if the contents of register $1=0$. If contents of register $1\neq0$, go to the next instruction.

312 Jump if contents of register $2=0$: conditional program transfer to the address (a) specified in the instruction register if the contents of register $2=0$. If the contents of register $2\neq0$, proceed to the next instruction.

313 Jump if contents of counter $P=0$: conditional program transfer to address (a) specified in the instruction register if the contents of counter $P=0$. If contents of counter $P\neq0$, proceed to the next instruction.

314 Jump if contents of carry storage$=1$: conditional program transfer to the address (a) specified in the instruction register if the contents of carry storage$=1$. If contents of carry storage$=0$, proceed to next instruction.

315 Jump if contents of borrow storage$=1$: conditional program transfer to address (a) specified in the instruction register if contents of borrow storage$=1$. If contents of borrow storage$=0$, proceed to the next instruction.

32x Jump if indicator latch Ix is in "one" condition: conditional program transfer to address (a) in the instruction register if the indicator latch is in condition "one." If indicator latch is in condition "zero" proceed to next instruction.

33x Set indicator latch x to "one" condition.

37x Reset indicator latch x to "zero" condition.

34x Execute plugboard controlled instruction x.

35x Sense on breakpoint switch x. The breakpoint switch is not shown in the circuit diagram. If breakpoint switch x is in "off" position, proceed to the next instruction. If the breakpoint switch x is in "transfer" position use the address (a) specified in the instruction register for the address of the next instruction. If the breakpoint switch x is in "stop" position, prepare to use the address (a) specified in the instruction register for the address of the next instruction and stop.

1957 western Joint

A Universal Computer Language Translator

R. B. BONNEY

NONMEMBER AIEE

CAN Univac talk to Bizmac? As more and more electronic data-processing systems are put into operation, the need for rapid and efficient information interchange becomes important. Many organizations have evolved data-processing systems using a combination of different computer types, together with a variety of input and output equipment. The use of different computers in a single organization does not necessarily represent poor planning, since the requirements of one group may dictate the use of a particular type of computer which is inadequate for the requirements of a different group.

To achieve efficient utilization of an electronic data-processing system, it is often desirable to transfer output data from one computer to a different computer for additional operations.

The need for interchange between different organizations is also beginning to develop, and can be expected to increase as electronic data-processing spreads. Acceptance of payroll deduction reports on International Business Machines Corporation (IBM) *705* magnetic tapes by the Treasury Department and the Social Security Agency are typical examples.

Transformation from one form of data to another is another common requirement. Conversion from magnetic tape to punched paper Teletype tape is representative of a wide variety of such requirements.

Unfortunately, the designers of electronic computers have made very little attempt at standardization. In nearly all cases, the recording format and coding used by the various computer manufacturers are sufficiently different to prevent direct interchange of data. Undoubtedly, the very rapid development and growth of the computer industry have been major factors in the lack of standardization.

The commonly used method of accomplishing data translation between computers up to now has been by means of punched cards, and in some cases punched paper tape. Translation by these methods is seriously hampered by the speeds of operation. Some special-purpose high-speed translation equipment has been built for direct translation of data on magnetic tape. However, because of the wide variation in requirements, the equipment developed for a specific application ordinarily cannot be used in a different application without modification or complete redesign.

Realization of the need for "universal" translation capability resulted from work on data-handling systems for scientific test data. For this purpose, a computer format control buffer was developed to accept unsynchronized, continuous digital input data which are to be converted to magnetic tape, in a format directly usable by a computer.

In discussing other applications of this equipment, it soon became evident that no two applications were alike, and that too much re-engineering would be required to adapt the equipment to each new situation. It was, therefore, decided that an attempt should be made to develop a system with the flexibility necessary to meet all commonly encountered translation requirements without becoming overcomplex and economically impractical. The Computer Language Translator is the result.

In considering all possible applications, there is a wide variety of requirements

---

R. B. BONNEY is with Electronic Engineering Company of California, Santa Ana, Calif.

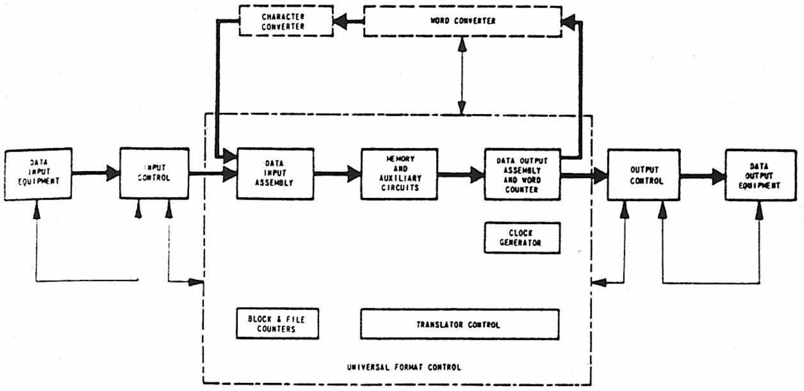


Fig. 1. Basic block diagram of Computer Language Translator (ZA-100)

which must be fulfilled by the translator equipment. A major undertaking in connection with this development has been the assembly of the myriad of details required for each computer format and for the numerous input and output devices which are expected to operate with the Translator. Although the equipment manufacturers have generally cooperated admirably, the problems in sorting out and verifying the necessary details have been considerable. The available literature is helpful but ordinarily does not give sufficient information, and it is difficult to locate the individual who can provide all of the answers.

Another problem has been to incorporate the necessary intelligence into the equipment without the Translator growing into a full-fledged digital computer.

The reader may ask, "Why not use one of the smaller general-purpose digital computers rather than develop new equipment specifically for translation?" There are a number of answers to this question:

1. All of the smaller computers currently available use magnetic drum memory, and as a result their speed of conversion would be much too low.

2. A magnetic drum is usually not suitable for delivering synchronous output data at a rate different from the input. This is one of the most fundamental translation requirements.

3. A large part of the translation process is involved with control of the data input and output devices. A general-purpose computer cannot provide these control features.

The required flexibility has been provided by means of the "building block" modular technique. Thus, by providing the proper combination of modules, the needs of a particular data processing system can be fulfilled without providing an overcomplicated system, containing excess features and equipment.

The design has been worked out around a foundation building block called the universal format control. This portion of the system is common to all applications. It contains a buffer memory, operator's controls, counters, and power supplies required for most operations.

Functions which are peculiar to a particular type of data input or output device are treated separately through the use of input and output control units. Ordinarily a separate control unit is required for each different type of data to be handled. This works out to be feasible from an equipment cost standpoint, since most input or output controls represent less than five % of the cost of the central universal format control.

Fig. 1 is a block diagram of the basic system. For multiple function systems, the universal format control can be shared between several input and output controls through the use of patchboards as shown in Fig. 2.

The operation of the Computer Language Translator can be divided into the two general categories of format conversion and code conversion.

Format conversion is concerned with the transformation of the physical characteristics of the input data to the required output form. Format conversion may involve radically different forms of input and output such as the translation of punched cards or punched tape to magnetic tape. On the other hand, the format conversion may require the translation of one magnetic tape to another magnetic tape with different recording speed and geometry.

Code conversion may or may not be required, depending on the application. Both character and word coding can be handled. Following the building block concept, the code converters are provided as auxiliary equipment when required, rather than as a part of the basic system.

In character conversion, the coding of each character is changed, as for example binary coded decimal to excess-three.

Word conversion involves reordering the characters within a word and transposition of the sign character where necessary.

Modification of the word order within a data block would be helpful in some cases. However, this feature has not been included in the interest of simplification. Deletion of certain words or characters from each block can be handled so long as the order remains fixed for any one operation.

In the normal operating mode, data is transferred from the input device to the buffer memory with the output equipment stopped. The amount of data entered can be arbitary, up to the capacity of the buffer memory (1,092 or 2,184 characters). On completion of loading, the input device is stopped, the output device is started, and the information is transferred from the buffer memory to the output.

In general, it is not practical to operate with the input and output devices running

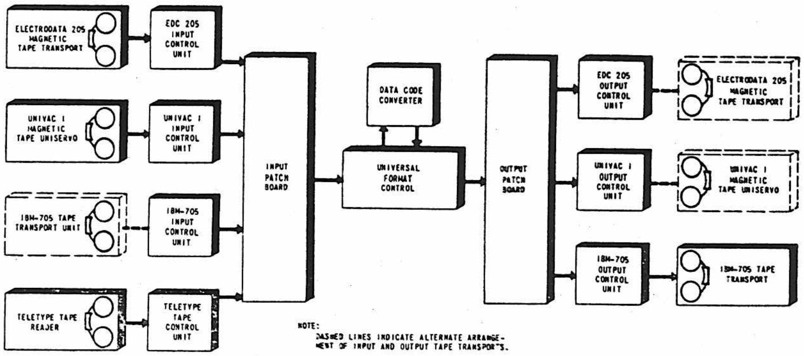

ELECTRODATA 205 MAGNETIC TAPE TRANSPORT
EDC 205 INPUT CONTROL UNIT
UNIVAC I MAGNETIC TAPE UNISERVO
UNIVAC I INPUT CONTROL UNIT
IBM-705 TAPE TRANSPORT UNIT
IBM-705 INPUT CONTROL UNIT
TELETYPE TAPE READER
TELETYPE TAPE CONTROL UNIT
INPUT PATCH BOARD
DATA CODE CONVERTER
UNIVERSAL FORMAT CONTROL
OUTPUT PATCH BOARD
EDC 205 OUTPUT CONTROL UNIT
ELECTRODATA 205 MAGNETIC TAPE TRANSPORT
UNIVAC I OUTPUT CONTROL UNIT
UNIVAC I MAGNETIC TAPE UNISERVO
IBM-705 OUTPUT CONTROL UNIT
IBM-705 TAPE TRANSPORT
NOTE:
DASHED LINES INDICATE ALTERNATE ARRANGEMENT OF INPUT AND OUTPUT TAPE TRANSPORTS.

Fig. 2. Typical multiple function Computer Language Translator system

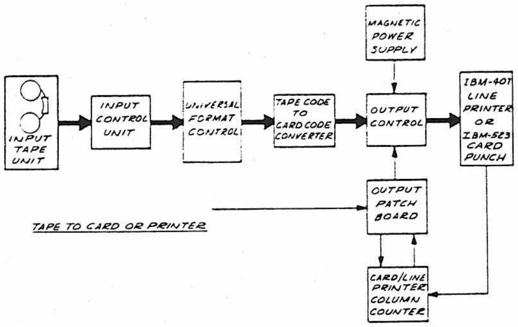


Fig. 3 (left). Translator system for magnetic tape to punched card or line printer

simultaneously. Unless the input and output rates are synchronized, the buffer memory will eventually overflow or empty. Since the input and output rates are normally different, the control logic is greatly simplified by using separate load and unload cycles.

In cases involving word conversion and for some other conversion operations, a recirculation cycle is introduced between the input and output cycles. This is illustrated in Fig. 1. On completion of transferring the input data to the memory, the recirculation cycle is initiated. During this cycle, the data is unloaded to the word converter one word at a time. The word converter is in turn unloaded, through the character converter, back into the buffer memory. At the completion of the recirculation cycle, all of the data will be in the required output coding. The memory can then be unloaded as a complete block at the rate required by the output device.

Insofar as possible, the equipment has been designed to eliminate the need for correlation of the input and output data formats.

The output word structure and block length can be entirely different from the word and block structure of the input data. In cases where fixed word length or block length output is required, zeros or other fill characters can be inserted into each word, and zeros or fill characters will automatically fill out to the proper block length. Several short input blocks can be compressed into a single output block.

Elimination of dependence between input and output format is also achieved by entering control characters into the memory. For example, when operating with variable input block length, an "end of data" character is entered into the buffer memory at the end of the input cycle. This special character is detected at the memory output and used to control the output cycle. The use of control characters thus eliminates the need for counting circuits to determine the output block length. The special characters, in most cases, do not appear in the output data.

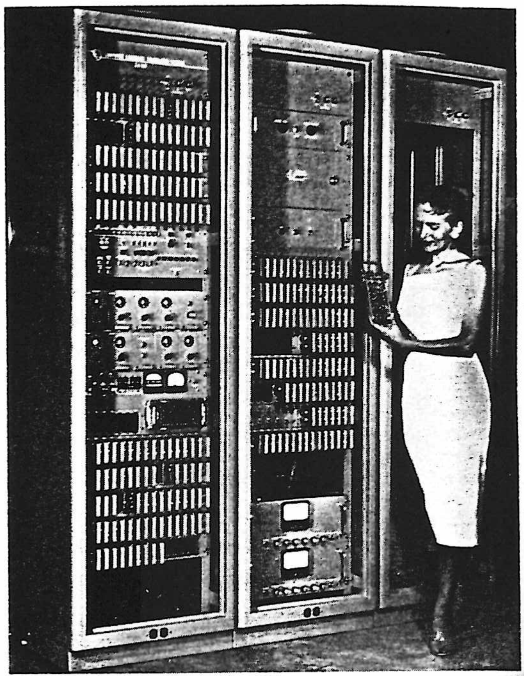


Fig. 4 (right). Computer - language Translator equipment

It is doubtful that a detailed functional block diagram of the Computer Language Translator would be of interest to most readers. The function of each component part may not be obvious from the block diagram without a step-by-step detailed description which is beyond the scope of this paper. It is felt rather that a brief listing of the functions and characteristics of the system will assist in illustrating the design philosophy.

As stated previously, all requirements peculiar to a particular input or output device are handled in the associated input or output control units. This results in much greater flexibility, since the designs for new types of control units can be completed individually without involving the majority of the system.

Input to the system can be in the form of: magnetic tape, punched paper tape, punched cards, analog to digital converter register, or other "on-line" source.

Output can be in the form of: magnetic tape, punched paper tape, punched cards, line printers, "on-line" registers for use by digital to analog converters, or other similar applications.

A rather formidable list of requirements for input and output control units results if all likely translator applications are considered. The situation is further complicated by various optional equipment combinations which may be desirable for various reasons. As an example, magnetic-tape data can be accepted directly from an IBM *727* tape unit, or *727* tapes can be played into the system using an Ampex *FR-307* or *FR-407* digital tape transport which is an integral part of the Translator. A somewhat different control unit is required for each case.

To date approximately 25 input and output control units have been designed or are in process. These include control units to handle magnetic tapes for IBM, ElectroData, and Remington Rand computers. Also included are control units for punched paper tape, IBM punched cards, and line printers.

By using the central portion of the system to fulfill more than one operational requirement, the cost per function decreases as additional operations are added. The central universal format control represents between 50 and 75% of the total equipment for most systems. The universal format control is designed to operate with any input or output control unit. Thus, the cost of adding an additional function to the system is essentially the cost of the input or output control unit plus any associated patchboard wiring.

To date Electronic Engineering Company has been reasonably successful in isolating the universal format control from special input or output data requirements. Anticipating the need for considerable flexibility, circuits needed to adapt the universal format control to different translation applications have been brought out to external terminals. By providing the necessary control signals to these terminals, any requirement within the basic system capability can be handled. In some cases, auxiliary equipment is designed to provide the necessary controls when they are not directly obtainable from the data input or output equipment.

All normal operator's controls are located on a panel housed in the universal format control. For magnetic tape input, provision is made for positioning the tape both automatically and manually in either forward or reverse direction. Indicating counters keep track of the block or file position of the tape.

The actual translation operation can also be controlled automatically or manually.

Other controls and indicators are associated with marginal testing and parity-error control circuits. All controls are interlocked to prevent improper operation.

Character parity is checked at the input before entering the buffer memory and also at the output of the memory. A third parity-check circuit is also provided for output parity checking when possible, as for example with the IBM *729* tape unit. At the operator's option, the controls can be set to automatically stop the operation if a parity error is detected.

Longitudinal (row) parity, if used, can also be checked by means of an optional module.

Marginal operation checks are provided for both the buffer memory and the d-c supplies for the balance of the system.

A common translation requirement is for off-line conversion from magnetic tape to punched cards or line printer and from punched cards to magnetic tape. The multipurpose features of the Computer Language Translator work to advantage here, since the basic system can be adapted to handle these off-line functions in addition to tape-to-tape or other requirements.

As anyone knows who has considered the problem, existing card-handling equipment and line printers are not easily adapted to magnetic tape. This is because the most commonly used equipment handles the data on a row-by-row rather than a column-by-column basis. Some column-by-column card-handling equipment is available, but it is too slow in most cases. A number of high-speed column-by-column readers are under development, but as yet are not generally available. The straight-forward approach to handling row-by-row cards is to provide a complete card size or printer line buffer memory which can be unloaded or loaded on either a row or column basis. Because of the number of parallel outputs involved (120 in the case of a line printer), this type of memory would be a relatively expensive addition to the translator system.

It is possible to utilize the buffer memory in the basic translator to handle cards and line printing by operating the memory in a manner equivalent to a magnetic drum. Using this approach, the stored information is recirculated in the same manner as discussed previously for word conversion.

Fig. 3 is a block diagram illustrating the method used for punching cards or controlling a line printer such as the IBM *407*.

Data corresponding to one card row or one line is first loaded into the memory from the input equipment. The memory is then recirculated 12 times in synchronism with the 12 card row positions. During each circulation cycle, a decoder detects characters which correspond to the row position and are to be punched or printed.

A column counter, and associated set-up patchboard, sets up the proper punch magnet or print magnet. Since this same counter controls unloading and reloading the memory, the required synchronization is maintained between the memory and the punch or printer columns. A similar approach is used for reading punched cards. Each row of the card is scanned sequentially by the column counter. A card-to-tape coder enters the proper character into the memory when a punch is detected.

Card reader verification can be handled by double reading. This necessitates a reader with two reading stations. Verification is done during the second reading by comparing the reader output against the decoded data entered into the memory during the first reading.

Card punching can be verified in a similar manner. In this case, the punched card is read following the punching cycle. The information from the reader is compared to the decoded output from the memory.

The translator is designed around a combination of plug-in vacuum tube circuits, diode logic circuits, transistor-driven ferrite cores, and in some cases tape wound cores. Some consideration was given to eliminating vacuum tubes entirely. However, the idea was not carried out because of the considerable amount of design knowledge and proven reliability of the vacuum tube plug-ins. These same components have been thoroughly proven by use in many other similar applications. On the other hand the transistor-driven ferrite core memory was used since this unit was originally developed using transistors. Tape wound cores are being used for the card and line printer column counter because of the relatively large linear count and number of output circuits involved.

In general, d-c logic is employed. Conservative design policies have been established in the interest of achieving high reliability. Active circuits are isolated from external loading. Definite rules for interconnected logic circuits are followed.

Standards are maintained for component parts and workmanship which are somewhat higher than are often found in commercial electronic equipment. Transformers, for example, are hermetically sealed. Other components are of top quality. It is sincerely believed, however, that the modest cost for high-quality equipment is more than offset by higher reliability and decreased maintenance requirements.

The basic system is housed in two equipment racks which are each 2 feet square and 7 feet high. One or more additional racks house input and output control units, patch-boards, auxiliary power supplies and, in some cases, magnetic tape transports.

Fig. 4 shows a typical translator system. This system was demonstrated at the Western Joint Computer Conference. The input to the system comes from punched paper Teletype tape or from ElectroData *205* magnetic tape. The output is in IBM *727* format. The input tape unit is the ElectroData transport used with the *205* Datatron computer. The output tape unit is an Ampex *FR-307* tape recorder with IBM head dimensions.

Until more varied application experience has been obtained, it will be difficult to assess the success or failure which has been achieved in developing a truly universal Computer Language Translator. Experience to date, based on actual applications and discussions with many potential users, has been very encouraging.